Méthodes probabilistes basées sur les mots visuels pour la reconnaissance de lieux sémantiques par un robot mobile

Mathieu Dubois

▶ **To cite this version:**

Mathieu Dubois. Méthodes probabilistes basées sur les mots visuels pour la reconnaissance de lieux sémantiques par un robot mobile. Robotique [cs.RO]. Université Paris Sud - Paris XI, 2012. Français. NNT : 2012PA112025 . tel-00679650

HAL Id: tel-00679650

https://theses.hal.science/tel-00679650

Submitted on 16 Mar 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ PARIS-SUD 11

ÉCOLE DOCTORALE : EDIPS

Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur

DISCIPLINE : Informatique

THÈSE DE DOCTORAT

soutenue le 20 février 2012

par

Mathieu Dubois

Méthodes probabilistes basées sur les mots visuels pour la reconnaissance de lieux sémantiques par un robot mobile

Directeur de thèse :	Philippe Tarroux	Professeur, ÉNS
Composition du jury :		
Président du jury :	François YVON	Professeur, Université Paris-Sud
Rapporteurs :	David FILLIAT	Professeur, ENSTA Paris Tech
	Philippe GAUSSIER	Professeur, Université de Cergy-Pontoise
Examinateurs :	Antoine CORNUÉJOLS	Professeur, Agro Paris Tech
	Christophe LEROUX	Ingénieur R&D, CEA

Yes, we can!

(B. Obama)

Nani gigantum humeris insidentes

(I. Newton parmi d'autres)

Remerciements

On écrit toujours ces quelques lignes en dernier et c'est bien dommage car tellement de gens mériteraient d'y figurer.

Je tiens à remercier mon directeur de thèse, Philippe Tarroux pour m'avoir soutenu durant ces années malgré mes doutes et mes impasses. Philippe a su rester humain et m'apporter beaucoup sur le plan de la science et des méthodes de travail.

Emmanuelle Frenoux mérite un merci tout particulier pour son enthousiasme a s'investir dans un domaine qui n'est pas le sien et son efficacité

Merci aux membres de mon jury, David Filliat, Philippe Gaussier, Antoine Cornuéjols, Christophe Leroux & François Yvon, pour leur intérêt dans mes modestes travaux et leurs questions pertinentes.

Il faudrait remercier ceux qui ont permis que la thèse se fasse, c'est-à-dire tout le LIMSI, quelques-uns des laboratoires voisins, les collègues enseignants, le personnel administratif, tous les copains *etc.* Allons-y en ordre.

Parmi tous, Hervé Guillaume mérite un merci tout particulier puisque j'ai repris la suite de ses travaux et qu'il a continué à m'aider plus de deux ans après sa soutenance. Je remercie les vénérables doctorants, Cédric et Sylvain Chevallier pour leur conseils précieux. Merci à Sonia qui a tout simplement contribué à ce que je reste en thèse, à Guillaume dont la générosité est malheureusement compensé par les goûts musicaux (je plaisante) et à Sylvain Caillou. Merci aux autres doctorants du groupe CPU. Merci aux stagiaires avec qui j'ai pu travailler et dont quelques-uns des travaux sont honteusement plagiés dans les pages suivantes : Bastien, Pèdre, Jean, Gautier, Mohammed, Jonathan & Christophe. J'espère avoir été un bon enseignant.

Que soient en particulier remerciés : Laurent Pointal pour ses compétences techniques et humaines, l'équipe AMIC du labo, la cellule expérimentale et les secrétariats (pour les bonbons et pour avoir réparé mes gaffes administratives). Merci à Anne Vilnat qui a mis une rage particulière à résoudre mes problèmes de financements. Enfin, personne au LIMSI ne peut travailler sans l'indispensable café de Nédée et les délicieux repas du CESFO.

Étant le dernier de ma promo au LIMSI, je remercie aussi tous les collègues de M2 (au LIMSI et ailleurs). Merci aux personnes que j'ai fréquenté aux cours de ces années et qui sont devenues des ami-e-s. Ils se reconnaîtront. Merci aux footballeurs du LIMSI qui m'ont permis de continuer à m'oxygéner.

Merci aux personnes qui ont organisé le pot de soutenance. Je suis le meilleur collègue de

bureau du monde et la quantité astronomique de cadeaux que j'ai reçu le prouve.

Merci bien sûr à ma famille qui m'a soutenu (« hé oui, je suis toujours étudiant ») et à mes amis de toujours l'être (ça m'étonne parfois). Merci à mes colocataires : Camille, Nicolas, Vincent & Boubacar. Big up aux membres d'un certain gang de la région bordelaise qui ont su perpétuer une amitié qui dure depuis oulà-ça-ne-nous-rajeunit-pas. Enfin, un gigantesque merci à Marina « Xoxo » Kugler pour tout.

Résumé

Les êtres humains définissent naturellement leur espace quotidien en unités discrètes. Ces unités, appelées « lieux sémantiques », sont caractérisées par une extension spatiale et une unité fonctionnelle. De plus, nous sommes capables d'identifier le lieu où nous sommes (*e.g.* le bureau 205) et sa catégorie (*i.e.* un bureau) sur la base de leur seule apparence visuelle. Les travaux récents en reconnaissance de lieux sémantiques visent à doter les robots de capacités similaires. Contrairement aux travaux en cartographie, ce problème est en général abordé comme un problème d'apprentissage supervisé.

Cette connaissance de nature sémantique peut être utilisée à plusieurs fins. Premièrement, le lieu courant définit le contexte d'action du robot, ce qui lui donne des indications sur la présence d'objets et leur apparence et contraint les actions possibles. Deuxièmement, ils fournissent une estimation qualitative de la localisation du robot. Cette information complémentaire aux techniques quantitatives, peut permettre au robot d'adapter ses déplacements. Enfin, cette connaissance de l'espace facilite le dialogue homme-machine.

Par rapports aux travaux de la littérature, cette thèse apporte deux principales nouveautés. Premièrement, nous combinons la caractérisation globale d'une image et les méthodes basées sur les mots visuels, qui reposent habituellement sur l'utilisation de descripteurs locaux. Les images sont décrites de manière globale afin de s'affranchir des variations de l'apparence des lieux. Nous testons plusieurs types de codages qui s'appuient sur des traits différents. Chaque mot visuel, obtenu par une phase de quantification vectorielle des signatures globales, représente donc une image, ce qui nous permet à la fois de réduire la variabilité des signatures et de simplifier l'apprentissage de modèles génératifs.

Deuxièmement, et de manière intimement reliée, la reconnaissance s'appuie sur plusieurs images, tirant parti du flux vidéo fourni par le robot, afin de diminuer l'ambiguïté liée à une image. Nous utilisons plusieurs méthodes d'intégration temporelle qui s'appuient sur un formalisme bayésien. Dans un premier modèle, nous ne tenons pas compte de l'ordre des images. Le mécanisme d'intégration est extrêmement simple mais montre des difficultés à repérer les changements de lieux. Nous élaborons donc plusieurs mécanismes de détection des transitions entre lieux qui ne nécessitent pas d'apprentissage supplémentaire. Une deuxième version enrichit le formalisme classique du filtrage bayésien en utilisant l'ordre local d'apparition des images.

Nous comparons nos méthodes à l'état de l'art sur des tâches de reconnaissance d'instances et de catégorisation, en utilisant plusieurs bases de données. Nous étudions en détail l'influence des paramètres sur les performances du système et nous comparons les différents types de codage employés sur une même base, ce qui était absent de la littérature. Ces expériences montrent que nos méthodes sont supérieures à l'état de l'art, en particulier sur les tâches de catégorisation, tout en restant simples.

Mots-clés : robotique autonome, reconnaissance de lieux sémantiques, filtrage bayésien, mots visuels, caractérisation globale des images

Abstract

Human beings naturally organize their space as composed of discrete units. Those units, called "semantic places", are characterized by their spatial extend and their functional unity. Moreover, we are able to quickly recognize a given place (e.g. office 205) and its category (i.e. an office), solely from their visual appearance. Recent works in semantic place recognition seek to endow the robot with similar capabilities. Contrary to classical localization and mapping work, this problem is usually tackled as a supervised learning problem.

Such a semantic knowledge can be used in several ways. First, the current place defines the context which gives clues about the presence and appearance of objects and constrains the set of possible actions. Second, a semantic place can be seen as a qualitative localization which complements classic localization techniques and may influence the navigation strategy followed by the robot. Finally, this semantic knowledge can be used to ease human-machine communication.

Our contributions are two-fold. First, we combine global images characterization and visual words methods which are usually based on local signatures. Global characterization is interesting because it is less sensitive to the appearance of a particular place. We compare several types of global characterization that rely on different properties. A visual word, obtained by quantizing global signatures, represents a whole image which reduce the noise and simplify learning of generative models.

Our second, but closely related, contribution is to use several images for recognition which allow to decrease uncertainty. We designed several Bayesian methods for temporal integration. Our first model doesn't use the natural temporal ordering of images. Temporal integration is very simple but has difficulties when the robot moves from one place to another. We thus develop several mechanisms to detect place transitions. Those mechanisms are simple and don't require additional learning. A second model augment the classical Bayesian filtering approach by using the local order among images.

We compare our methods to state-of-the-art algorithms on place recognition and place categorization tasks. We study the influence of system parameters and compare the different global characterization methods on the same dataset. These experiments show that our approach, while being simple, leads to better results especially on the place categorization task.

Keywords: autonomous robotics, semantic place recognition, bayesian filtering, visual words, global image characterization

Table des matières

Introduction 1

I. Robotique et vision 5
1. Robotique cognitive . . . 5
   1.1. Définition et objectifs . . . 5
   1.2. Navigation . . . 6
   1.3. Différentes approches en robotique cognitive . . . 13
   1.4. Approche probabiliste du SLAM . . . 17
   1.5. Conclusion . . . 22
2. Vision par ordinateur . . . 23
   2.1. Description phénoménologique des images naturelles . . . 26
   2.2. Caractérisation globale des images . . . 35
   2.3. Extraction de zones d'intérêt . . . 40
   2.4. Caractérisation locale des images . . . 46
   2.5. Conclusion . . . 47
3. Apprentissage artificiel . . . 48
   3.1. Introduction . . . 48
   3.2. Apprentissage génératif et discriminatif . . . 51
   3.3. Conclusion . . . 57
4. Conclusion générale . . . 58

II. État de l'art en reconnaissance de scènes 69
1. Reconnaissance visuelle . . . 69
   1.1. Introduction . . . 70
   1.2. Reconnaissance d'objets . . . 73
   1.3. Reconnaissance de scènes . . . 76
   1.4. Vue d'ensemble des techniques utilisées en reconnaissance de scènes . . . 81
2. Approches basées sur les caractéristiques bas-niveau . . . 82
   2.1. Approches globales . . . 82
   2.2. Approches locales . . . 85
   2.3. Approches statistiques . . . 86
   2.4. Conclusion . . . 88

3. Approches basées sur les mots visuels . . . 88
3.1. Principe de base . . . 89
3.2. Avantages et inconvénients . . . 95
3.3. Améliorations . . . 96
4. Approches basées sur la modélisation sémantique . . . 99
4.1. Modélisation sémantique des régions . . . 100
4.2. Reconnaissance préalable des objets . . . 101
4.3. Utilisation des propriétés perceptuelles . . . 104
5. Conclusion . . . 106

III. Reconnaissance de lieux sémantiques en robotique 119
1. Introduction . . . 119
1.1. Le concept de lieu sémantique . . . 119
1.2. Reconnaissance visuelle de lieux sémantiques . . . 120
1.3. Intérêt pour la robotique . . . 121
2. Position du problème . . . 122
2.1. Différences par rapport au SLAM . . . 122
2.2. Différences par rapport à la reconnaissance de scènes . . . 123
2.3. Position formelle du problème . . . 124
3. Challenges . . . 126
3.1. Variabilité des lieux . . . 126
3.2. Difficultés liées aux images . . . 130
3.3. Extension spatiale d'un lieu et continuité temporelle . . . 131
3.4. Conclusion . . . 133
4. État de l'art en reconnaissance de lieux sémantiques . . . 134
4.1. Vue d'ensemble de l'état de l'art . . . 134
4.2. Formats de description de l'image . . . 138
4.3. Classification et confiance dans la décision . . . 139
4.4. Intégration temporelle . . . 141
5. Conclusion . . . 145

IV. Reconnaissance de lieux avec des modèles sac de mots 153
1. Introduction . . . 153
2. Codages globaux et catégories sémantiques . . . 154
3. Architecture du système . . . 157
3.1. Vue d'ensemble . . . 157
3.2. Signatures utilisées . . . 159
3.3. Entraînement des cartes de Kohonen . . . 161
3.4. Apprentissage et intégration temporelle . . . 161

4. Résultats en reconnaissance d'instances . . . 166
4.1. Protocole . . . 166
4.2. Influence de la longueur d'intégration . . . 167
4.3. Influence de la taille de la carte de Kohonen . . . 167
4.4. Robustesse aux variations de luminosité . . . 173
4.5. Comparaison avec l'état de l'art et conclusion . . . 174
5. Résultats en catégorisation . . . 177
5.1. Protocole . . . 177
5.2. Influence de la longueur d'intégration . . . 177
5.3. Influence de la taille de la carte de Kohonen . . . 180
5.4. Comparaison avec l'état de l'art et conclusion . . . 180
5.5. Conclusion . . . 185
6. Conclusion . . . 186

V. Détection visuelle des changements de lieux **189**
1. Introduction . . . 189
2. Comportement au voisinage des seuils . . . 190
2.1. Comportement moyen de l'entropie autour des transitions . . . 192
3. Présentation des algorithmes . . . 195
3.1. Seuillage de l'entropie . . . 195
3.2. Détection des maximas locaux de l'entropie . . . 199
3.3. Utilisation des changements de classification . . . 199
4. Évaluation . . . 199
4.1. Résultats en reconnaissance d'instances . . . 201
4.2. Résultat en catégorisation . . . 205
5. Conclusion . . . 211

VI. Utilisation de l'ordre local pour la reconnaissance de lieux **215**
1. Introduction . . . 215
2. Filtrage bayésien sur chaînes de Markov . . . 216
2.1. Présentation générale du modèle . . . 216
2.2. Algorithme de filtrage bayésien . . . 217
2.3. Apprentissage des paramètres . . . 218
3. Résultats en reconnaissance d'instances . . . 219
3.1. Influence de la matrice de transition . . . 220
3.2. Influence de la taille de la carte de Kohonen . . . 220
3.3. Comparaison au filtrage bayésien classique . . . 220
3.4. Détection des transitions . . . 223
3.5. Comparaison avec l'état de l'art et conclusion . . . 225

4. Résultats en catégorisation . . . 225
4.1. Influence de la matrice de transition . . . 225
4.2. Influence de la taille de la carte de Kohonen . . . 226
4.3. Comparaison au filtrage bayésien classique . . . 226
4.4. Détection des transitions . . . 228
4.5. Comparaison à l'état de l'art et conclusion . . . 228
5. Conclusion . . . 229

VII Conclusion et perspectives **235**
1. Contributions . . . 235
2. Perspectives . . . 237
2.1. Améliorations du système . . . 237
2.2. Travaux futurs . . . 238

A. Bases de données utilisées **243**
1. Base de données COLD . . . 243
1.1. Environnements . . . 243
1.2. Procédure d'acquisition . . . 246
1.3. Problèmes connus . . . 247
1.4. Protocole de reconnaissance d'instances de lieux . . . 247
1.5. Protocole de catégorisation . . . 248
2. Base de données VPC . . . 248
2.1. Environnements . . . 248
2.2. Procédure d'acquisition . . . 250
2.3. Protocole de catégorisation . . . 250
3. Stockage des données . . . 251

B. Cartes auto-organisatrices **255**
1. Principe . . . 255
1.1. Carte et cellules . . . 255
1.2. Apprentissage compétitif et projection des données . . . 255
2. Description détaillée . . . 256
2.1. Algorithme d'apprentissage . . . 256
2.2. Voisinage . . . 258
3. Paramètres et ensemble d'apprentissage . . . 260
3.1. Paramètres . . . 260
3.2. Ensemble d'apprentissage . . . 261

C. Fonctions de texture **263**
1. Fonctions de texture sur un histogramme du premier ordre . . . 263

2. Fonctions de texture sur un histogramme du second ordre 263
3. Signatures basées sur les fonctions de textures 264

D. Théorie de la détection **267**
1. Cas de la classification binaire 267
1.1. Table de contingence 267
1.2. L'espace ROC 269
1.3. Courbe ROC 270
2. Cas d'un problème à plus de deux classes 271

Table des figures

I.1. Schéma des interactions d'un robot avec l'environnement. 7
I.2. Différents types de cartes 10
I.3. Exemple de système de cartographie sémantique 12
I.4. Contrôleur hiérarchique 14
I.5. Contrôleur réactif 16
I.6. Réseau bayésien décrivant l'évolution de l'état du monde et des variables observées par le robot. 18
I.7. Modèle de déplacement basé sur l'odométrie. 21
I.8. Une image et sa représentation informatique. 24
I.9. Exemple de figure ambiguë. 25
I.10. Exemple d'images artificielles et naturelles. 27
I.11. Exemple de distributions et de leur kurtosis sur une échelle logarithmique. . 32
I.12. Base de fonctions optimale déterminée par [OF96]. 33
I.13. Partie réelle et transformée de Fourier du filtre de Gabor. 34
I.14. Incorporation de la géométrie dans la description des images. 36
I.15. Descripteur GIST. 37
I.16. Descripteur CENTRIST. 39
I.17. Transformations géométriques d'une image. 43
I.18. Le modèle de Itti, Koch et Niebur [IKN98]. 45
I.19. Descripteur SIFT. 47
I.20. Apprentissage discriminatif et génératif. 55

II.1. Exemples et contre-exemples de scènes. 71
II.2. Schéma de principe de la reconnaissance visuelle. 73
II.3. Sources de variations rencontrées en reconnaissance d'objets. 75
II.4. Exemple de catégorie perceptuelle et de catégorie fonctionnelle. 76
II.5. Propriétés statistiques des scènes naturelles. Tiré de [TO03]. 79
II.6. Illustration du problème de multi-classification. 80
II.7. Importance de l'information globale et locale pour la catégorisation de scènes. 83
II.8. Utilisation de l'histogramme d'orientation pour discriminer les scènes d'intérieur et les scènes d'extérieur. 84
II.9. Application de l'ACI aux catégories d'images naturelles. 87

II.10. Illustration de l'approche sac de mots. 89
II.11. Principe de l'algorithme de clustering de [NS06]. 92
II.12. Exemple d'image où un grand nombre de mots d'arrière-plan sont détectés. . 96
II.13. Illustration de deux approches incorporant des informations géométriques dans les techniques Bag-of-Words (BoW). 98
II.14. Approche utilisant la classification sémantique de régions. 101
II.15. Illustration des approches basées sur la forme. 102
II.16. Illustration des approches locales basées sur les parties d'objets. 104

III.1. Différence entre le concept de lieu dans le SLAM topologique et le concept de lieu sémantique. 123
III.2. Scénario pour la reconnaissance de lieux sémantiques. 125
III.3. Exemple de l'influence de la luminosité ambiante sur l'apparence d'un même lieu. 127
III.4. Exemple de l'influence de l'activité des occupants sur l'apparence d'un lieu. . 128
III.5. Exemple des ressemblances entre des pièces de la même catégorie dans un environnement donné. 128
III.6. Exemples de variabilité entre des pièces issues de la même catégorie fonctionnelle. 129
III.7. Les objets exhibent des configurations spatiales très différentes dans des lieux d'une même catégorie. 129
III.8. Exemples d'occultation du lieu. 130
III.9. Exemples d'images portant peu d'information. 131
III.10 Différents vues d'un même lieu peuvent exhiber de grandes différences. . . . 132
III.11 Différences entre la reconnaissance de scènes et la reconnaissance de lieux sémantiques. 133
III.12 Résultats en reconnaissance d'instances de Ullah et al. 135
III.13 Résultats en catégorisation de Ullah et al. 137
III.14 Techniques d'intégration de Mozos et Pronobis. 144

IV.1. Clustering effectué sur des environnements d'intérieur structurés. 156
IV.2. Architecture du système de reconnaissance de lieux. 158
IV.3. Influence de L en reconnaissance d'instances. 168
IV.4. Influence de S en reconnaissance d'instances. 169
IV.5. Influence de S sur l'apprentissage. 171
IV.6. Images associées à la cellule la plus vraisemblable de la classe couloir. 172
IV.7. Résultats en reconnaissance d'instances obtenus avec le descripteur CENTRIST ($S = 50$, $L = +\infty$). 175
IV.8. Résultats en reconnaissance d'instances obtenus avec le descripteur GIST sur les canaux de couleurs ($S = 30$, $L = +\infty$). 176
IV.9. Influence de L en catégorisation. 178

IV.10 Résultats obtenus en catégorisation avec les fonctions de texture. 179
IV.11 Influence de S en catégorisation. 181
IV.12 Exemples de résultats en catégorisation avec le système tBoW. 183

V.2. Comportement des fenêtres d'intégration temporelle lorsque l'on n'utilise pas la notification des changements de lieux. 191
V.3. Utilisation de fenêtres de tailles différentes à l'instant t. 193
V.4. Entropie moyenne 20 pas de temps après la transition pour différentes valeurs de ℓ. La barre verticale montre la transition. 194
V.5. Utilisation d'une fenêtre de taille fixe pour la détection de transitions. 194
V.6. Évolution de l'entropie autour d'une transition à $t = 0$ et en dehors des transitions pour $\ell = 30$. 196
V.7. Lissage de l'entropie à l'aide d'une moyenne pondérée exponentiellement. . . 197
V.8. Filtrage des transitions détectées. 198
V.9. Évaluation de la détection des seuils. 200
V.10. Distribution de l'écart entre deux transitions détectées. 210

VI.1. Réseau bayésien décrivant notre modèle. 216
VI.2. Influence de p_e en reconnaissance d'instances. 221
VI.3. Influence de S en reconnaissance d'instances. 222
VI.4. Influence de p_e en catégorisation. 226
VI.5. Influence de S en catégorisation. 227
VI.6. Exemples de résultats en catégorisation avec BFMC. 230

A.1. Cartes des différents environnements utilisés pour la base COLD. 244
A.1. Cartes des différents environnements utilisés pour la base COLD (suite). . . 245
A.2. Schéma de la base de données utilisée dans nos expériences. 253

B.1. Illustration de l'entraînement d'une SOM. 257
B.2. Différentes fonctions de distance et les voisinages de taille 1 associés. 259
B.3. Disposition des cellules dans l'espace 2D. 259
B.4. Différentes formes de la carte. 260

D.1. Espace ROC et courbes ROC. 270

Liste des tableaux

III.1. Résultats en catégorisation obtenus par Wu et Ranganathan. 137

IV.1. Nombre de dimensions du descripteur GIST avant et après projection. 160
IV.2. Valeurs de S utilisées dans la Section IV.4.3. 167
IV.3. Distance d'intersection moyenne entre les distribution d'apprentissage et les distributions en reconnaissance (pour la tâche de reconnaissance d'instances). 172
IV.4. Robustesse aux variations de luminance en reconnaissance d'instances. 173
IV.5. Distance d'intersection moyenne entre les distribution d'apprentissage et les distributions en reconnaissance. Plus la distance d'intersection est proche de 1, plus les distributions sont similaires. 182
IV.6. Performances obtenues sur la base VPC pour le descripteur CENTRIST. . . 184
IV.7. Performances obtenues sur la base VPC pour le descripteur GIST. 185

V.1. Valeur de ℓ utilisées. 193
V.2. Résultats de la détection des seuils pour le protocole de reconnaissance d'instances avec le descripteur CENTRIST. 203
V.3. Résultats de la détection des seuils pour le protocole de reconnaissance d'instances avec le descripteur GIST. 204
V.4. Résultats de la détection des seuils pour le protocole de catégorisation sur la base COLD avec le descripteur CENTRIST. 206
V.5. Résultats de la détection des seuils pour le protocole de catégorisation sur la base COLD avec le descripteur GIST. 207
V.6. Résultats de la détection des seuils pour le protocole de catégorisation sur la base VPC avec le descripteur CENTRIST. 208
V.7. Résultats de la détection des seuils pour le protocole de catégorisation sur la base VPC avec le descripteur GIST. 209

VI.1. Comparaison du filtrage bayésien et de Bayesian Filtering with Markov Chains (BFMC) pour le descripteur CENTRIST 223
VI.2. Comparaison du filtrage bayésien et de BFMC pour le descripteur GIST (luminance) 224
VI.3. Comparaison du filtrage bayésien et de BFMC pour le descripteur GIST (couleur) 224
VI.4. Comparaison du filtrage bayésien et de BFMC pour la détection des transitions. 224

VI.5. Résultats moyens pour le protocole de catégorisation sans filtrage, avec le filtrage bayésien classique et avec BFMC. 227

VI.6. Comparaison du filtrage bayésien et de BFMC pour la détection des transitions en catégorisation. 228

VI.7. Performances obtenues sur la base Visual Place Categorization (VPC) pour le descripteur CENTRIST. 231

VI.8. Performances obtenues sur la base VPC pour le descripteur GIST. 231

VI.9. Performances obtenues sur la base VPC pour le descripteur GIST. 232

A.1. Catégories utilisées dans la base COLD. 247

A.2. Nombre de séquences pour chaque environnement, chemin et condition d'illumination dans la base COLD. 247

A.3. Conditions expérimentales pour le protocole de reconnaissance d'instances. . 249

A.4. Catégories utilisées dans la base VPC. 250

A.5. Nombre minimum et maximum d'images par classe dans les échantillons d'apprentissage et de reconnaissance pour le protocole Section A.2.3. 251

D.1. Table de contingence et définitions associées. 268

D.2. Problème multiclasse. 271

Introduction

Objectifs

Après des années de recherche, le projet de voir des robots nous aidant dans la vie quotidienne commence à se réaliser : les premiers robots de service sont déjà disponibles et on s'attend à ce que ce marché connaisse une forte croissance dans les années à venir. De plus, des robots à caractère éducatif et ludique sont aussi disponibles. De tels robots sont amenés à interagir avec des êtres humains dans des environnements complexes et dynamiques et doivent donc être dotés d'une certaine autonomie.

Parmi les capacités de base nécessaires à la création d'un robot autonome, la navigation et les problématiques reliées, comme la localisation et la cartographie, sont certainement les plus étudiées. Il existe aujourd'hui de nombreuses techniques permettant à un robot de construire une carte détaillée de son environnement et de naviguer à l'aide de cette carte. La plupart du temps, ces cartes représentent juste les parties de l'environnement accessibles au robot, sans information sur la nature du lieu.

Néanmoins, dans la vie de tous les jours, les êtres humains arrivent à se localiser à l'intérieur d'un bâtiment sans carte. Bien plus, des connaissances de haut-niveau leur permettent de choisir des actions en fonction du lieu où il se trouvent.

Au-delà de la navigation, il est donc nécessaire de doter les robots d'une représentation de l'espace de plus haut niveau, compatible [1] avec celle de l'être humain. Ce domaine de recherche est appelé cartographie *sémantique*. Une première étape consiste à être capable d'identifier la nature de l'environnement dans lequel se trouve le robot : Est-ce un environnement d'extérieur ou d'intérieur ? Est-ce un bureau, un couloir ou une chambre ? Répondre à ces questions constitue le but de la reconnaissance de lieux sémantiques, domaine auquel nos travaux se rattachent.

On peut distinguer deux grands problèmes :

1. reconnaître un lieu précis, déjà connu, par exemple le bureau dans lequel j'ai effectué ma thèse ; cette problématique est appelée reconnaissance d'instances de lieux.
2. reconnaître la catégorie d'un lieu éventuellement jamais vu, par exemple reconnaître la cuisine dans un appartement où nous ne sommes jamais allés ; on parle alors de catégorisation de lieux.

1. Nous choisissons ce mot pour signifier que la représentation artificielle n'a pas à être identique ou à être basée sur les même processus que la représentation chez l'être humain.

Reconnaître un lieu de façon à pouvoir lui associer une étiquette sémantique est utile à trois points de vue. Premièrement, le fait pour le robot de partager certains concepts de haut-niveau avec les humains peut faciliter la communication homme-machine. Deuxièmement, en fonction du lieu dans lequel il se trouve, le robot est susceptible de faire telle ou telle action. La connaissance du lieu courant va donc influencer la sélection de l'action. Troisièmement, certains objets apparaissent préférentiellement dans certains lieux. Connaissant le contexte d'évolution du robot, il est possible d'orienter un processus de reconnaissance d'objets vers les classes d'objets les plus probables et leurs caractéristiques (position et apparence).

Les techniques de cartographie reposent de manière cruciale sur l'intégration des données proprioceptives et extéroceptives. Au contraire, la reconnaissance de lieux s'appuie essentiellement sur des données extéroceptives. Parmi les différentes modalités, la vision semble la plus adaptée à la reconnaissance de lieux sémantiques par un robot mobile, ne serait-ce que parce que la vision semble être la principale modalité utilisée par les humains. Par rapport aux capteurs lasers traditionnellement utilisés en cartographie, elle fournit des informations plus riches.

Les lieux sémantiques peuvent exhiber une grande diversité d'apparence. De plus, celle-ci peut changer au cours du temps, sur des échelles allant de quelques heures à plusieurs mois. C'est pourquoi la question de la robustesse des méthodes de reconnaissance de lieux est un des enjeux majeurs du domaine.

Le domaine de la reconnaissance de lieux sémantiques est relativement récent en robotique mais de nombreux travaux ont été effectués ces dernières années. Citons par exemple le projet européen CoSy (*Cognitive Systems for Cognitive Assistants*, voir [Chr+09]). Des bases de données spécialisées ont récemment vu le jour. Dans ce contexte, une activité de reconnaissance visuelle de lieux sémantiques a été initiée au LIMSI. Nos recherches font suite aux travaux de ce groupe, notamment ceux de Hervé Guillaume (voir [Gui09]), qui ont montré qu'on peut reconnaître des lieux en utilisant une caractérisation fruste de l'image, à condition de combiner les informations extraites de plusieurs images.

L'objectif de nos travaux est, d'une part, de développer des méthodes de reconnaissance de lieux basées sur les travaux précédents et, d'autre part, d'évaluer leur robustesse.

Organisation du manuscrit

Le chapitre I présente des généralités sur les liens entre vision par ordinateur, robotique cognitive et apprentissage artificiel. Nous verrons que ces trois domaines font un recours massif aux outils probabilistes. En effet, ceux-ci permettent de modéliser naturellement l'incertitude dont sont affectés les capteurs, les effecteurs et les connaissances d'un robot.

Le chapitre II présente un domaine spécifique de la vision par ordinateur, la reconnaissance de scènes. Bien que les deux problèmes soient distincts, de nombreux travaux en reconnais-

sance de lieux sémantiques, dont les nôtres, utilisent des techniques développées pour la reconnaissance de scènes. Il est donc intéressant de faire un état de l'art des techniques dans ce domaine, notamment sur les questions de représentation de l'image.

Le chapitre III présente en détail la problématique de la reconnaissance de lieux sémantiques. Nous expliquons les différences entre la reconnaissance de lieux sémantiques, la reconnaissance de scènes et les approches classiques de localisation topologique. Nous présentons ensuite les difficultés de la reconnaissance de lieux sémantiques puis nous présentons l'état de l'art dans ce domaine. Nous insistons en particulier sur l'importance de l'évaluation des systèmes obtenus.

Le chapitre IV présente le système issu des travaux de Hervé Guillaume. Nous évaluons le système sur des bases de données plus représentatives de la diversité des lieux sémantiques et nous étudions l'influence de plusieurs paramètres, ce qui nous permet de mettre en lumière certaines propriétés du système. Enfin, ce chapitre, permettra de comparer les représentations de l'image proposées par différents auteurs. Globalement, les résultats du système sont supérieurs à l'état de l'art mais reposent sur un composant externe chargé de détecter les changements de lieux.

Le chapitre V présente quelques travaux destinés à munir notre système d'un tel composant. Contrairement à la plupart des approches proposées dans la littérature, nous utilisons la modalité visuelle : la détection des changements de lieux est effectuée, sans apprentissage supplémentaire, sur la base de résultats de la classification. Les méthodes présentées sont très simples mais donnent des résultats intéressants.

Le chapitre VI présente une évolution plus récente de nos travaux qui vise à dépasser certaines hypothèses des travaux précédents. Ces travaux enrichissent le filtrage bayésien en ajoutant des corrélations entre les observations. Évalué sur les mêmes bases de données, ce système montre des performances supérieures à l'état de l'art.

Enfin, le dernier chapitre résume nos contributions et propose des pistes d'amélioration et des perspectives pour de futurs travaux.

Références

[Chr+09] H. Christiensen et al., éds. *Cognitive Systems*. 2009 (cf. p. 2).

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Département d'informatique, juil. 2009 (cf. p. 2).

I. Robotique et vision

Le but de ce premier chapitre est de situer notre travail dans un cadre plus large et de donner les fondements sur lesquels nous nous appuierons dans la suite. Nous présenterons succinctement les trois grands domaines auxquels ce travail se rattache : la robotique cognitive, la vision par ordinateur et l'apprentissage artificiel. Ces deux derniers domaines seront présentés en tenant compte des spécificités du domaine robotique.

1. Robotique cognitive

Cette section vise à introduire le domaine de la robotique cognitive. Nous illustrerons en particulier ses liens avec la perception, l'action et l'apprentissage au travers d'une des problématiques centrales : la navigation.

1.1. Définition et objectifs

1.1.1. Robot autonome mobile

Le terme « robot » possède plusieurs significations. Le type de robot le plus répandu est sans doute le robot industriel effectuant des tâches répétitives dans les usines. D'après le rapport de l'International Federation of Robotics (IFR), il y avait en 2009 entre 1 million et 1,3 million de robots industriels en activité dans le monde. Ce secteur d'activité représentait en 2009 un chiffre d'affaire de 19 milliards de dollars. Selon la définition retenue par l'ISO, un robot industriel est un manipulateur polyvalent, contrôlé automatiquement et reprogrammable, ayant trois axes ou plus. Cette définition ne fait pas référence à un environnement ou à des capacités cognitives.

Il est cependant naturel de vouloir réaliser un robot doté de capacités de réflexion. Bien que ce souhait ait été présent dès la construction des premiers automates, les premiers programmes de recherche en robotique cognitive sont apparus dans les années 60 en lien avec l'intelligence artificielle. Dans le cadre de ce manuscrit, nous définissons la robotique cognitive comme la branche de la robotique visant à doter un robot de capacités de perception, de décision et d'action qui lui permettent d'agir sur son environnement de manière autonome et de réagir seul à des situations imprévues [1].

1. Un robot autonome n'est donc pas un esclave, contrairement à l'étymologie du mot. Rappelons que le

Une des premières capacités que l'on souhaite implémenter dans un robot mobile est la capacité de se déplacer dans l'environnement. On parle alors de robot autonome mobile.

1.1.2. Architecture de contrôle

Un robot autonome est muni de capteurs pour acquérir de l'information sur le monde environnant et d'effecteurs pour agir sur ce monde. De manière générale, le robot va effectuer des opérations sur ses perceptions et ses actions (voir Figure I.1). On nomme « représentations internes » ou « modèle du monde », la manière dont sont représentées les perceptions et les actions. Suivant l'approche adoptée, ces représentations peuvent être très simples, voire inexistantes (par exemple, le robot peut juste stocker la valeur courante des capteurs et des effecteurs) ou au contraire très élaborées (par exemple, le robot peut construire un modèle détaillé du monde en 3D).

La manière dont sont agencés ces trois composants s'appelle l'architecture de contrôle. Il existe une grande variété d'architectures de contrôle mais on peut en général les classer en deux grandes familles (voir Section I.1.3) : les approches hiérarchiques, qui utilisent des représentations internes élaborées, et les approches réactives, dans lesquelles les représentations sont très simples (certaines approches refusant même l'usage de représentations). Il existe aussi de nombreux travaux qui se basent sur une architecture mixte. La réalisation particulière d'une architecture de contrôle et son implantation dans un robot est appelée un contrôleur.

1.2. Navigation

De manière très générale, la navigation est la tâche qui consiste à s'approcher d'un but. C'est donc une des tâches les plus importantes pour un robot mobile (voir [LL90 ; Tru+97 ; FM00 ; Fil09]). Nous présentons dans cette partie les sources d'information utilisées pour la navigation (voir Section I.1.2.1) puis les différentes stratégies utilisées pour aborder le problème (voir Section I.1.2.2) et les différents types de cartes (voir Section I.1.2.3). Enfin, nous présentons le problème classique de la construction d'une carte et de la localisation simultanées, dans lequel le robot doit en même temps cartographier son environnement et se localiser dans cette carte.

Nous rencontrerons dans cette partie le concept de lieu. Précisons cependant que les lieux auxquels nous ferons référence ne sont pas des lieux sémantiques : ce sont des positions dans

terme est apparu dans la pièce « R.U.R (Rossum's Universal Robots) » de Karel Capek, publiée en 1920. Il vient du tchèque « robota » qui désigne la corvée au sens médiéval du terme puis au sens actuel du terme, ce qui montre bien que les robots sont conçus comme des serviteurs dociles et efficaces pour réaliser les tâches pénibles. Notons que dans la pièce originale déjà, les robots se révoltent (et possèdent donc une certaine autonomie). Ce thème de la révolte des travailleurs dociles est déjà présent dans le mythe juif du Golem. Notons enfin que le mythe de travailleurs mécaniques était aussi présent en Chine, dans les légendes nordiques et chez les Grecs.

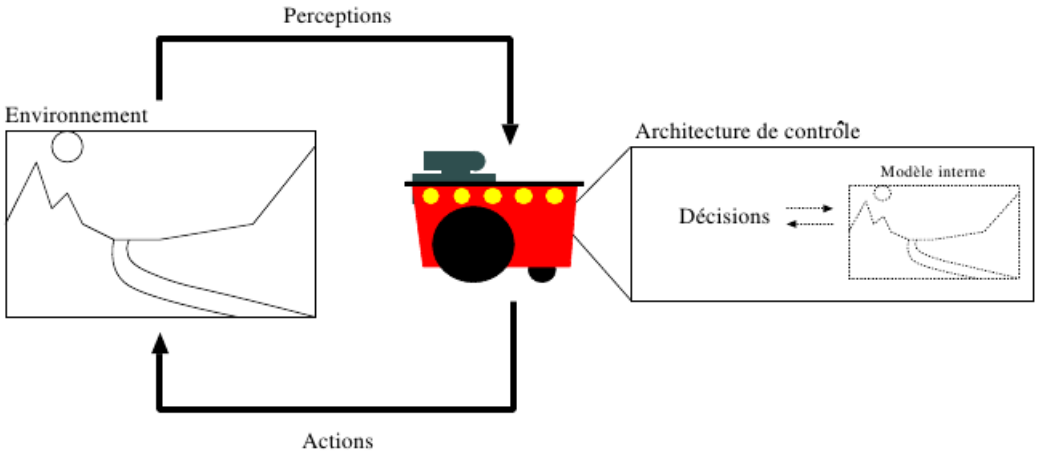


FIGURE I.1.: Schéma des interactions d'un robot avec l'environnement. Le modèle interne de l'environnement est optionnel. Extrait de [Fil09].

l'espace alors qu'un lieu sémantique est défini par son extension.

1.2.1. Sources d'informations pour la navigation

Un robot dispose de deux sources d'information principales pour se localiser et naviguer :
- Les informations proprioceptives (ou idiothétiques) qui sont fournies par des capteurs internes au robot et qui le renseignent sur son déplacement dans l'espace.
- Les informations extéroceptives (ou allothétiques) ou plus simplement les perceptions qui sont des informations à propos de l'environnement et qui sont caractéristiques d'une position. Ces informations peuvent être de nature très variée. Par exemple, un robot peut mesurer la distance à laquelle se trouvent des obstacles avec des capteurs infrarouges ou utiliser une caméra.

Informations proprioceptives Il existe deux types de capteurs proprioceptifs qui peuvent renseigner un robot sur son déplacement :

1. les capteurs mesurant la rotation de ses roues (ces capteurs sont appelés « capteurs odométriques ») ;
2. les centrales inertielles qui mesurent l'accélération du robot.

En accumulant ces informations au cours du temps, on peut estimer la position relative de deux points grâce à un processus d'intégration.

Ces informations ont l'avantage de ne pas être perturbées par l'environnement extérieur et de fournir directement la position du robot dans un repère euclidien. En revanche, les erreurs à chaque pas de temps sont accumulées et l'estimation de la position devient de moins en

moins fiable au cours du temps. Il est notoire en robotique que les informations d'odométrie sont entachées d'une erreur importante et que l'estimation de la position du robot sur la base de ces seules données n'est pas fiable.

Informations extéroceptives Les informations extéroceptives renseignent le robot sur son environnement. Il existe plusieurs types de capteurs extéroceptifs :

- capteurs infrarouges ;
- télémètres lasers ou à ultrasons ;
- caméras (simples ou panoramiques) et éventuellement système de vision stéréoscopique.

Ces informations permettent au robot de se positionner dans l'espace indépendamment de ses déplacements et donc ne sont pas soumises à l'accumulation d'erreurs.

1.2.2. Stratégies de navigation

Trullier et al. [Tru+97] distinguent deux grands types de stratégies de navigation que nous allons détailler : les stratégies sans modèle interne et les stratégies avec modèle interne.

Parmi les stratégies sans modèle interne, on distingue :

1. L'approche d'un objet par des actions réflexes. L'exemple le plus connu est sans doute les véhicules de Braitenberg (voir [Bra84]) capables de s'orienter vers une source de lumière. Ces véhicules réalisent simplement une remontée ou une descente de gradient sur l'intensité de la lumière. Ils correspondent à un simple contrôleur proportionnel en automatique et sont donc sujets à des oscillations dans le comportement du robot.
2. La stratégie de guidage par des amers qui permet d'atteindre un lieu défini par une configuration spatiale d'objets remarquables ; cette stratégie utilise une descente de gradient pour reproduire la configuration mémorisée. Ce type de stratégie est par exemple utilisé par les abeilles.

Ces stratégies permettent de naviguer vers un point sous condition que le but ou les amers décrivant le but soient perceptibles.

Les stratégies basées sur un modèle interne sont :

3. La stratégie d'action associée à un lieu (*recognition-triggered response*) : dans cette stratégie, à chaque lieu est associée une action permettant de rejoindre le but ; lorsqu'un lieu est reconnu, l'action associée est déclenchée ; le robot va donc suivre des routes fixées qui le mènent vers un but ;
4. La navigation topologique : le modèle interne est un graphe qui mémorise des lieux (définis comme un ensemble de zones de l'espace dans lesquelles les perceptions sont similaires) et les relations d'accessibilité entre ces lieux ; le robot est ainsi capable de composer un nouveau chemin à partir de segments connus ; cependant, le robot ne peut pas atteindre des endroits qu'il n'a pas encore visités ou emprunter des chemins qu'il n'a pas déjà parcourus.

5. La navigation métrique : le modèle interne stocke les positions dans un repère commun ; le robot peut donc calculer les positions relatives entre deux lieux, ce qui permet de calculer des déplacements même s'ils n'ont pas été mémorisés sous forme d'un lien.

Les stratégies 1 à 3 sont basées sur des actions réflexes et ne nécessitent pas une représentation globale du monde. On les regroupe sous le terme de navigation réactive. Les stratégies 4 et 5 nécessitent la construction d'une carte *i.e.* d'un modèle détaillé du monde. L'utilisation d'une carte n'est pas réservée aux êtres humains. En effet, de nombreux auteurs ont montré l'existence de cellules spécialisées dans la représentation de l'espace chez les mammifères (voir [OC78]) formant de véritables cartes cognitives (voir [ON78 ; GC96]) [2].

Nous détaillons dans la section suivante les différents types de cartes utilisées en robotique.

1.2.3. Les différents types de cartes

Suivant [TBF05], nous définissons une carte comme une collection « d'objets », pris au sens très général : $m = \{m_1, m_2, \ldots, m_N\}$.

On distingue généralement deux types de cartes selon la nature des informations spatiales qu'elles incorporent : les cartes métriques et les cartes topologiques. Ces cartes ne doivent pas être confondues avec les stratégies de navigation définies à la page 8.

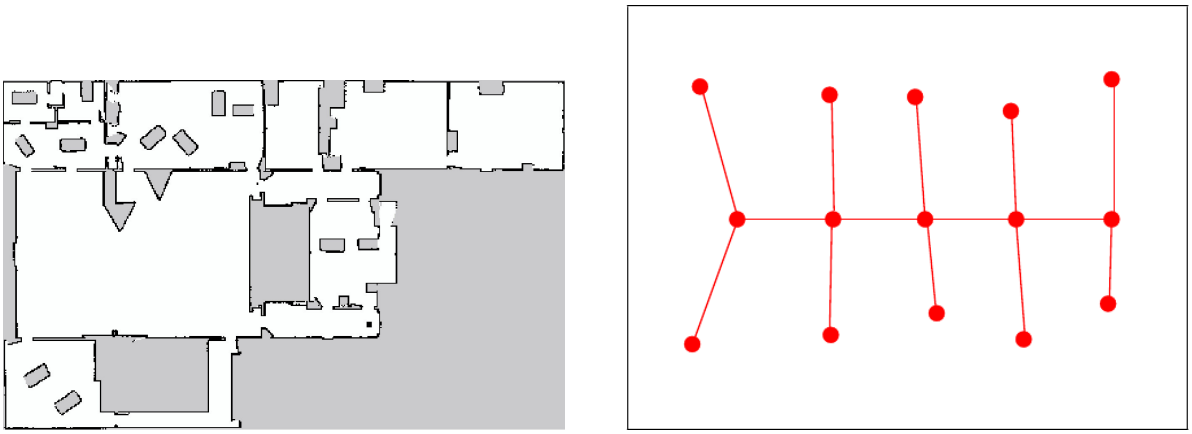
Une définition un peu plus précise consiste à parler de cartes basées sur les coordonnées (*location-based maps*) et de cartes basées sur les points de repères (*feature-based maps*) :

- Dans les cartes basées sur les coordonnées, chaque position est étiquetée. Autrement dit, l'indice n correspond à la position (x, y) dans un repère absolu ; ce sont ces cartes que l'on désigne sous le nom de cartes métriques.
- Dans les cartes basées sur les points de repère, l'objet m_n contient la position et les propriétés d'un point de repère particulier (*landmark* ou *feature*) ; ces cartes sont désignées sous le nom de cartes topologiques. Contrairement à ce que le nom pourrait faire penser, les cartes topologiques contiennent aussi la position des points dans un repère global (ce qui est indispensable pour les stratégies de navigation évoluées)

Ces deux types de cartes se différencient par la manière de discrétiser l'environnement. Dans les approches basées sur les coordonnées, l'environnement est découpé densément en cellules, en général de taille fixe. Un exemple classique d'une telle représentation est la grille d'occupation (voir figure I.2a) dans laquelle chaque cellule est représentée par une variable binaire dont la valeur code l'occupation (0=libre, 1=occupé). Le but de la cartographie est donc de déterminer, sur la base des mesures, la probabilité pour une cellule d'être libre.

Les cartes topologiques reposent sur une discrétisation plus grossière de l'environnement (voir figure I.2b), ce qui les rend plus efficaces d'un point de vue computationnel : seuls des

2. L'hypothèse de l'existence de telles cartes, évoquée pour la première fois par Tolman [Tol48], est d'ailleurs une des points de départ des sciences cognitives.



(a) Grille d'occupation (b) Carte topologique

FIGURE I.2.: Différents types de cartes

endroits remarquables, qui servent de point de repère, sont stockés. Ces points de repères forment les nœuds de la carte topologique. Il existe plusieurs manières de choisir les points de repères ([Fil09]) :

- Nœuds définis par le concepteur : dans ces approches, le concepteur définit directement quels points de repère doivent être détectés par le robot et les procédures permettant de détecter ces zones. Les positions les plus couramment utilisées sont les couloirs, les portes et les intersections. Une variante extrême est l'utilisation de balises (*beacon*) spécialement conçues pour une détection et une identification robuste.
- Nœuds définis à des positions canoniques : dans ces approches, le concepteur dote le robot de capacités pour détecter des lieux intéressants (les portes) mais la détection et l'identification ont lieu lors de l'exploration de l'environnement.
- Nœuds définis de manière non-supervisée : dans ces approches, la définition des nœuds n'est pas fixée à l'avance. Pour mettre en œuvre une telle approche, il faut définir un critère qui permette de décider quand un nouveau lieu a été atteint. L'approche la plus simple consiste à créer un nouveau nœud quand le robot a parcouru une certaine distance. Une autre approche consiste à créer un nouveau nœud lorsque la distance entre l'entrée perceptuelle courante et le dernier nœud visité excède un certain seuil. Enfin, des approches bayésiennes permettent au robot de sélectionner les points de repère les plus informatifs (voir [Thr98a ; RD09]).

Les approches métriques ont pour avantage d'être précises et de représenter l'ensemble de l'espace de manière continue. Cependant, quelle que soit l'approche adoptée, l'espace métrique de la carte est construit à partir de la position du robot, estimée à partir des données proprioceptives, et de la distance des obstacles, estimée à partir des entrées extéroceptives (voir Section I.1.4). Si l'estimation sur la position du robot est inexacte, cette erreur sera

propagée au reste de la carte et notamment aux coordonnées estimées des objets. De plus, l'espace étant continu, cette erreur est propagée de proche en proche, jusqu'à ce que la dérive rende la carte inutilisable. Bien que des méthodes probabilistes récentes (voir Section I.1.4) dépassent pour partie ces limitations, la cartographie métrique d'environnement vaste reste difficile en raison des dérives d'estimation de position.

Les approches topologiques sont en général plus simples à mettre en œuvre et sont efficaces pour la navigation (la recherche d'un chemin se réduit à une recherche dans un graphe). De plus, comme nous l'avons vu, les nœuds sont parfois définis par le concepteur en fonction de caractéristiques significatives pour un être humain telles que les portes, les intersections, *etc.* La communication homme-machine s'en trouve facilitée. Enfin, les nœuds sont plus faciles à ancrer dans les perceptions du robot. Les approches topologiques nécessitent en revanche souvent une exploration plus complète de l'environnement.

Plus récemment, les méthodes de cartographie dite *sémantique* sont apparues ([Gal+05 ; NH08]). Ces approches intègrent ces types de cartes et contiennent en plus des informations de haut niveau, par exemple la classe et la position de certains objets, les caractéristiques et fonctions de ces objets, *etc.* Le but est de permettre au robot de raisonner avec des concepts proches de ceux de l'être humain (voir Figure I.3). Pour cela, des connaissances de haut-niveau sont ajoutées dans le système, par exemple au moyen d'ontologies.

Pour plus de détails et de références bibliographiques, nous renvoyons le lecteur au cours de Filliat (voir [Fil09, chap .3]) et à la thèse de Pronobis (voir [Pro11]).

1.2.4. Le problème du SLAM

Pour les environnements simples, il est possible de fournir une carte au robot, par exemple à partir des plans du bâtiment. Cependant, ces plans ne sont pas toujours précis et ne répertorient pas les divers objets à l'intérieur du bâtiment qui peuvent gêner la progression du robot (et dont la position peut varier au cours du temps). De plus, ces cartes ne sont pas toujours disponibles (qu'on songe par exemple à des robots devant opérer après une catastrophe naturelle). Il est donc nécessaire pour le robot de cartographier son environnement en même temps qu'il l'explore. Ce problème est connu sous le nom de « cartographie et localisation simultanées » (en anglais, *Simultaneous Location And Mapping (SLAM)*).

Ce problème est difficile car la construction d'une carte implique de localiser les différents objets par rapport à soi, ce qui suppose de se localiser, donc d'avoir une carte. Le problème du SLAM semble donc se ramener à un problème du type « l'œuf ou la poule ? ».

Nous étudierons dans la Section I.1.4 les approches probabilistes de ce problème. La section suivante est consacrée à l'étude des grands paradigmes de la robotique et à leurs applications en navigation.

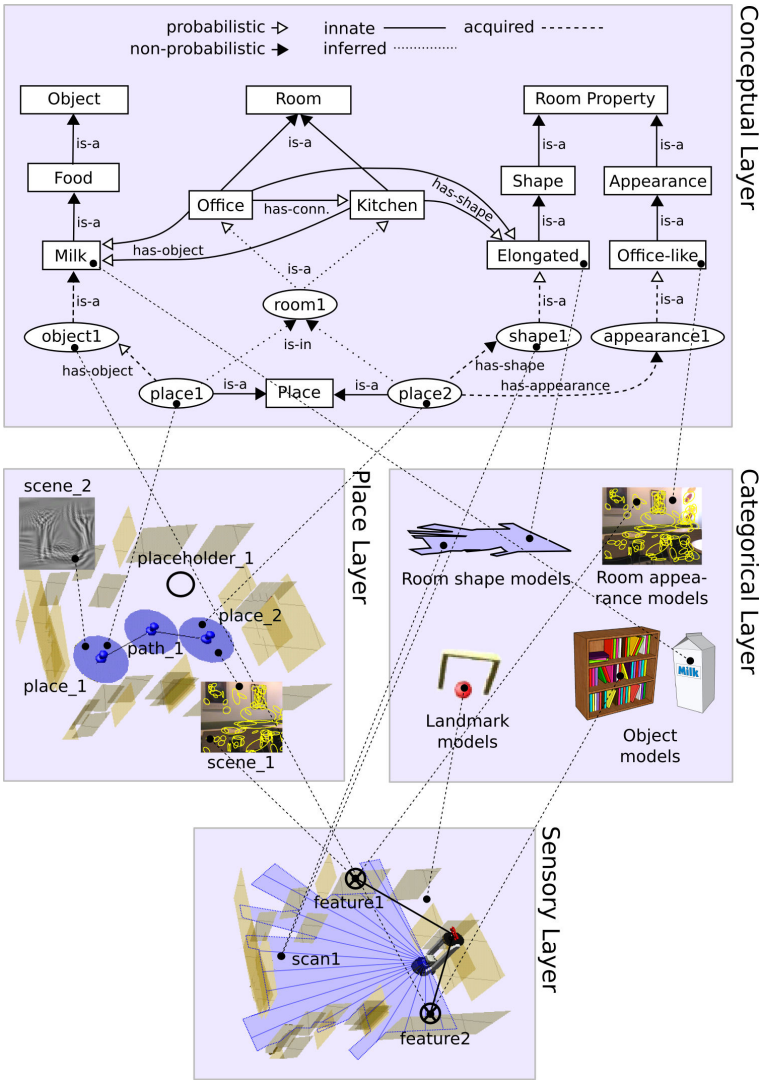


FIGURE I.3.: Exemple de système de cartographie sémantique. Le système est composé de quatre couches correspondant à différents niveaux d'abstraction. La première couche contient les entrées sensorielles et représente l'environnement immédiat du robot. La deuxième couche est constituée de la carte des lieux (*Place layer*), qui contient une représentation topologique de l'environnement, et de la carte des catégories (*Categorical layer*) qui contient une catégorisation de l'environnement. Au-dessus, la couche conceptuelle contient une représentation unifiée de l'environnement dans un système de représentation des connaissances basé sur les ontologies. Extrait de [Pro11].

1.3. Différentes approches en robotique cognitive

Nous présentons ici les deux grandes familles d'approches qui correspondent peu ou prou au développement historique de la discipline. Cette classification n'est bien sûr pas définitive puisque les travaux développés dans une approche sont repris dans une autre et que de nombreux travaux utilisent des techniques issues de plusieurs paradigmes. Le lecteur au fait de ces évolutions peut passer directement à la Section I.1.4.

1.3.1. Approche symbolique

Historiquement, la robotique cognitive était une branche de l'intelligence artificielle et les premières approches partageaient donc les mêmes hypothèses fondatrices, issues des travaux en sciences cognitives des années 50, eux-mêmes dans la lignée des travaux fondateurs de Fregge, Turing, *etc.*

L'hypothèse principale de ces travaux est que la cognition (assimilée au cerveau) est un système de manipulation de symboles, de la même manière qu'un mathématicien manipule des symboles abstraits. Les règles de manipulation de ces symboles sont syntaxiques, c'est-à-dire qu'elles ne dépendent pas du sens des symboles. Cette hypothèse est connue sous le nom d'hypothèse computo-représentationnelle. La notion de représentation interne y joue un rôle central.

La cognition est donc réduite à un système de traitement de l'information : les entrées sont traitées par une unité de traitement (*processor*) pour donner des sorties.

Ces modèles découpent la cognition en sous-tâches étudiées indépendamment : représentation du savoir (*knowledge representation*), planification, *etc.* Ces sous-tâches sont étudiées sous l'angle de la résolution de problème (*problem solving*).

L'architecture de contrôle typique est représentée dans la figure I.4. Dans ce type de contrôleur, un composant central sert à planifier les actions en se basant uniquement sur un modèle interne du monde. Dans cette approche la perception sert uniquement à construire le modèle interne. On peut donc dire que dans ce modèle la perception est passive : les caractéristiques du monde sont là, on doit juste les acquérir [3]. Cette approche est souvent appelée *Sense-Model-Plan-Act*, délibérative ou encore descendante (*top-down*). Le modèle interne est supposé décrire complètement le monde. Cette hypothèse est appelée « hypothèse du monde clos ». Cette approche est appelée hiérarchique car la résolution d'un problème fait appel à des sous-problèmes, eux-mêmes décomposés jusqu'à atteindre des actions élémentaires (voir Steels [Ste90]). L'exécution d'un plan va ainsi générer un ensemble d'actions.

Un des succès les plus marquants de cette approche a été le robot « Blocks World » (voir [Win72]) qui est capable de manipuler des cubes de couleurs dans un environnement simplifié. Un opérateur peut lui transmettre des ordres simples (impliquant un cube à la fois)

3. On pourra citer la célèbre anecdote du professeur qui avait fixé pour but comme travail d'été à ces étudiants de résoudre le problème de la vision (voir [Bod06, p. 781]).

en langage naturel comme : déplacer le bloc bleu au-dessus du bloc vert. Si le bloc n'est pas accessible, le robot refuse d'exécuter le mouvement.

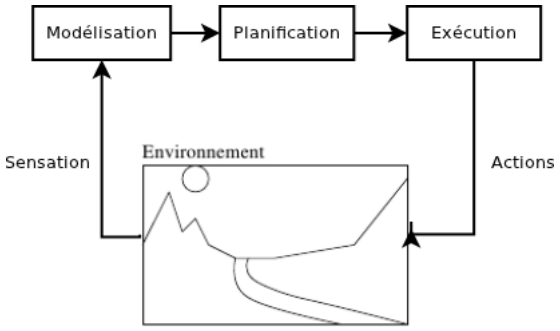


FIGURE I.4.: Contrôleur hiérarchique

Cette approche a été utilisée pour la navigation par exemple dans le robot Shakey (voir [Nil84] [4]). Shakey est capable de se déplacer dans un environnement simple, composé de surfaces planes peintes uniformément et décrit par un système symbolique de logique du premier ordre. Le monde est réparti en cinq classes : les portes, les murs, les pièces, les objets et le robot. Certains points disposent d'un signe distinctif permettant de les identifier. La localisation précise se base sur un système d'analyse de scène (voir Section II.4.1) : un algorithme de segmentation basique permet de découper l'image en régions et les frontières entre régions sont détectées. Un système de règle permet de classer les régions dans l'une des cinq classes en tenant compte des contraintes (par exemple le sol ne peut pas se trouver au-dessus des murs). Enfin, la connaissance d'un point distinctif et d'une ligne permet de connaître la position et l'orientation du robot. Des routines de vision bas-niveau permettent de localiser des objets simples composés de faces planes et peintes de couleurs différentes. Shakey peut aussi reconnaître des prises (identifiables facilement car elles sont noires sur fond blanc) pour s'y recharger et des interrupteurs pour allumer la lumière. Un opérateur humain peut lui indiquer un but via un ordinateur distant.

Les limites de cette approche sont liées à l'hypothèse d'un monde clos décrit par un modèle interne. Il est facile de faire un tel modèle pour un monde « jouet » statique. En revanche, cette hypothèse s'est avérée inadaptée dans les situations réelles pour au moins trois raisons :

1. Le temps de calcul nécessaire pour intégrer les perceptions au modèle et planifier une action peut s'avérer très long (plusieurs dizaines de minutes sur Shakey) ; dans un environnement dynamique, l'action choisie peut finalement se révéler non-pertinente.
2. Les perceptions sont souvent bruitées, ambiguës voire absentes. Ces erreurs se retrouvent dans le modèle interne qui ne correspond plus au monde ; les décisions prises seront

4. Ces travaux, réalisés de 1966 à 1972, abordaient déjà la plupart des problèmes de robotique et de vision par ordinateur.

inadaptées. Par exemple dans Shakey, l'algorithme de segmentation regroupe d'abord les pixels de valeur proche dans une région puis les régions similaires sont regroupées. Ce type d'approche, déjà faillible dans l'environnement de Shakey, ne produit pas des résultats robustes dans les environnements réels.

3. Enfin, la construction d'un modèle pour les environnements réels et les tâches compliquées peut s'avérer impossible.

Une autre limite est liée à la perception passive de l'environnement. De nombreuses études (voir section I.2) insistent sur le fait que la perception n'est pas juste une étape passive. Au contraire, pour faire face à la complexité et à l'ambiguïté des sensations, la perception doit être vue comme une étape d'inférence. D'un certain point de vue, dans l'approche symbolique, la perception n'existe pas.

On peut noter que de tels systèmes ne font en général pas appel à un apprentissage.

1.3.2. Approche située

En réponse à ces robots lents, incapables d'agir dans le monde réel, Brooks théorisa une nouvelle approche dans une série de papiers influents ([Bro90 ; Bro91b ; Bro91a]). Cette nouvelle approche embrasse les travaux de Brooks (voir [Bro86]) et rétrospectivement un certain nombre de travaux pionniers (voir [Bra84] et les tortues cybernétiques de Grey Walter). Cette approche met l'accent sur la réalisation concrète de robots plus petits et plus simples mais capables d'agir rapidement dans le monde réel, complexe et dynamique (on parle de robotique située).

Les échecs de l'approche symbolique montrent que l'interaction avec le monde réel est plus difficile que le raisonnement symbolique. L'un des problèmes principaux vient du fait que les symboles du système ne sont pas reliés à la perception du monde réel mais sont définis par le concepteur et peuvent difficilement être extraits des données. Littéralement, le robot ne sait pas de quoi il parle. Ce problème est connu sous le nom de problème de l'ancrage des symboles (*symbol-grounding problem*). Il est intéressant de noter que, d'après Brooks, les approches situées sont nées du besoin des ingénieurs d'avoir des robots fonctionnant dans le monde réel. Cependant, ces travaux ont profondément modifié la manière d'envisager la cognition et l'intelligence artificielle, notamment le rôle des représentation et la notion de comportement (ces travaux ont été désignés sous le terme de « nouvelle IA »).

Différentes approches ont été proposées dans ce paradigme (voir [Mat02 ; PS99]) : l'approche réactive pure (qui inclut l'architecture de subsomption et les approches par champ de potentiel), dans laquelle il n'y a pas de représentation du tout, l'approche comportementale, dans laquelle le robot utilise des représentations distribuées par exemple dans un réseau de neurones, l'approche bio-mimétique. Elles partagent néanmoins certains traits caractéristiques :

- l'importance d'une boucle sensori-motrice courte afin d'agir rapidement ;
- l'émergence de comportements, c'est-à-dire de structures d'interaction entre le robot et l'environnement ; ces activités émergentes ne sont pas codées dans le système ;

– l'importance de l'inspiration biologique (poussée à son paroxysme dans l'approche biomimétique).

Un exemple d'architecture utilisée est l'architecture de subsomption proposée par Brooks [Bro86]. Ici, le contrôleur est constitué de modules faiblement couplés (voir figure I.5). Chaque module prend en entrée les données sensorielles et délivre en sortie un ordre moteur. Les modules étant faiblement couplés, chacun d'eux réalise une action élémentaire (par exemple tourner, avancer, *etc.*). De plus, les modules sont organisés en couches, chaque couche réalisant une compétence (par exemple l'évitement d'obstacles, le suivi d'un mur, *etc.*). Enfin, les modules d'une couche peuvent subsumer (c'est-à-dire « prendre pour prémisses ») les couches inférieures. L'architecture de subsomption a été qualifiée de découpage vertical du problème en raison de l'empilement de couches, par opposition au découpage horizontal de la Figure I.4.

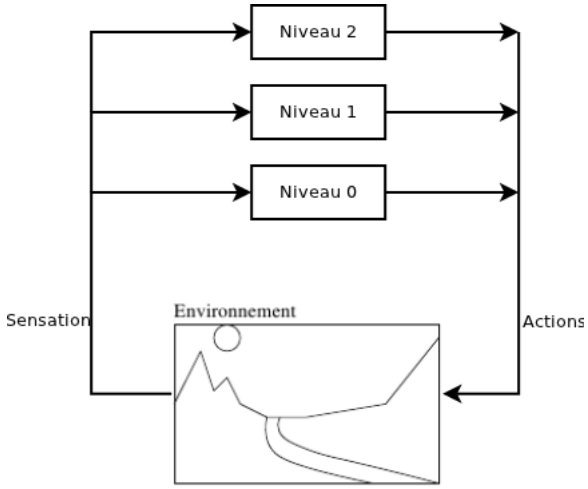


Figure I.5.: Contrôleur réactif

Parmi les premiers travaux utilisant cette approche, [Tho+88 ; Tur+88] ont développé le premier robot capable de conduire sur une route (en réponse au projet *Autonomous Land Vehicle* du DARPA). De nombreux travaux en navigation utilisent cette approche (par exemple, [CQG07 ; Arl+00 ; FM00 ; AMF99]). Mataric [Mat90] a montré comment on pouvait construire des cartes à partir de cette approche. Les stratégies de navigation sont, en général, des actions associées à un but ou des stratégies de navigation topologique.

Cette approche a permis de grand progrès en matière de navigation, notamment pour les stratégies 1 à 3. Les autres stratégies, qui font appel à une représentation globale de l'environnement, se sont révélées plus problématiques. Par exemple, dans [Mat90], le robot peut naviguer dans un environnement fermé (les bureaux du MIT) en suivant les murs. La navigation et la construction d'une carte dans un espace ouvert se révèlent beaucoup plus difficiles. De plus, les cartes obtenues sont difficilement interprétables par un humain ou alors

liées à quelques buts précis. Par exemple dans [CQG07], le robot navigue entre trois lieux correspondant à trois besoins vitaux (se nourrir, boire, se reposer), suivant son besoin. De plus, la construction de cartes repose de manière cruciale sur des informations d'orientation absolue, fournies par un compas.

De manière générale, les approches situées nécessitent de disposer de données sensorielles précises. En effet, toute erreur dans les capteurs se transforme rapidement en erreur dans les sorties du module, ce qui peut fausser le contrôle. De plus, elles se sont parfois avérées complexes à mettre en œuvre (voir [HP91]).

Certaines approches situées utilisent des techniques d'apprentissage, principalement des techniques issues des réseaux de neurones, de l'apprentissage non-supervisé et de l'apprentissage par renforcement (voir [PS99, chap. 14]).

Un certain nombre de travaux ont exploré les architectures hybrides pour essayer de combiner le meilleur des deux mondes : les niveaux supérieurs font appel à des modèles du monde pour raisonner et le contrôle bas-niveau est assuré par un contrôleur réactif. Nous ne développerons pas davantage ces méthodes.

1.4. Approche probabiliste du SLAM

Les techniques probabilistes ([Thr01 ; TBF05]) ont été introduites en robotique dans le sillage de l'approche située. Cette approche se base sur des modèles statistiques des capteurs et des effecteurs. De même, les modèles internes sont probabilistes. Ces modèles intègrent donc naturellement l'incertitude dont sont affectés les capteurs et les effecteurs et l'aspect stochastique du monde extérieur. Ainsi, dans un problème de navigation, au lieu de donner une seule estimation de la localisation, les algorithmes probabilistes se basent sur la distribution de probabilité de toutes les localisations. De plus, ces modèles sont capables d'intégrer plusieurs sources d'information (voir Section I.1.2.1), car ces sources sont représentées de manière unifiée sous forme de probabilité.

Cette approche s'est imposée grâce à sa robustesse dans le monde réel et au fait qu'on dispose d'algorithmes efficaces. De ce point de vue, on peut dire qu'elle augmente le caractère situé de la robotique. Notons cependant qu'il n'y a en général qu'un seul contrôleur monolithique. De plus, d'un point de vue philosophique, les probabilités sont une forme de représentation.

En terme de navigation, ces modèles utilisent une carte et donc les stratégies de navigation topologique ou métrique.

Nous présentons tout d'abord les grands principes des méthodes probabilistes avant de parler de leur apport au problème du SLAM.

1.4.1. Interaction robot-environnement

Le formalisme probabiliste permet d'intégrer les mesures et les actions du robot au cours du temps de manière efficace. Plus formellement, les croyances du robot sur son état interne

et l'état du monde au temps t sont représentées par une variable aléatoire $\boldsymbol{Y}_t$ [5]. Cette variable représente tout ce qui peut influer sur l'évolution du monde. Dans un problème de localisation, cette variable contiendra les coordonnées (x, y) et l'orientation θ du robot dans un repère global. Dans un problème de SLAM, cette variable contiendra aussi la carte m (voir Section I.1.4). De même, les actions et les mesures effectuées par le robot au temps t sont représentées par une variable aléatoire ($\boldsymbol{U}_t$ pour les actions, $\boldsymbol{X}_t$ pour les mesures). Le robot doit estimer la séquence des état cachés $\boldsymbol{Y}_{1:t}$. Pour cela, il connaît uniquement la séquence des observations, $\boldsymbol{X}_{1:t}$, et des actions qu'il a effectuées, $\boldsymbol{U}_{1:t}$. Les croyances (*beliefs*) du robot au temps t sont représentées par la densité de probabilité $\mathrm{p}\left(\boldsymbol{Y}_t|\boldsymbol{X}_{1:t}, \boldsymbol{U}_{1:t}\right)$, qu'on note $\mathrm{bel}\left(\boldsymbol{Y}_t\right)$ [6].

L'approche probabiliste repose sur les deux hypothèses suivantes :

1. L'hypothèse markovienne de l'évolution de l'état stipule que la connaissance de l'état au temps t dépend uniquement de l'état précédent : $\mathrm{p}\left(\boldsymbol{Y}_t|\boldsymbol{Y}_{0:t-1}, \boldsymbol{X}_{1:t-1}, \boldsymbol{U}_{1:t}\right) = \mathrm{p}\left(\boldsymbol{Y}_t|\boldsymbol{Y}_{t-1}, \boldsymbol{U}_t\right)$. Cette distribution de probabilité est appelée modèle de transition car elle représente l'évolution de l'état du monde en fonction des actions du robot (notons cependant qu'elle dépend aussi de la dynamique propre du monde).
2. L'hypothèse de l'état complet stipule que l'état courant explique complètement l'observation courante : $\mathrm{p}\left(\boldsymbol{X}_t|\boldsymbol{Y}_{0:t}, \boldsymbol{X}_{1:t-1}, \boldsymbol{U}_{1:t}\right) = \mathrm{p}\left(\boldsymbol{X}_t|\boldsymbol{Y}_t\right)$. De plus, cette distribution de probabilité est souvent indépendante de t : $\mathrm{p}\left(\boldsymbol{X}_t|\boldsymbol{Y}_t\right) = \mathrm{p}\left(\boldsymbol{X}|\boldsymbol{Y}\right)$. On parle alors de modèle de capteur bien que la réponse du modèle dépend de l'état du monde.

Les relations entre les variables sont représentées graphiquement par le réseau bayésien de la figure I.6. Ce formalisme entretient une relation profonde avec les modèles de Markov cachés.

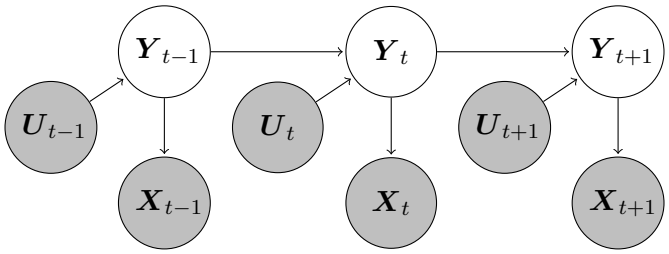


FIGURE I.6.: Réseau bayésien décrivant l'évolution de l'état du monde et des variables observées par le robot. Les nœuds grisés sont les variables observables. Les nœuds clairs sont les variables cachées (inconnues).

5. Les lettres en gras représentent des quantités vectorielles. Les variables aléatoires sont notées en majuscules et leur valeur observée en minuscules.

6. Il est d'usage en robotique de noter $\boldsymbol{x}$ les croyances du robot et $\boldsymbol{z}$, les observations. Nous avons choisi une convention différente pour harmoniser les notations avec le domaine de l'apprentissage où $\boldsymbol{x}$ représente en général la donnée et y, la classe à inférer.

1.4.2. Estimation récursive de l'état et filtrage bayésien

Le but de l'estimation récursive de l'état est d'exprimer $\text{bel}(\boldsymbol{Y}_t)$ en fonction de $\text{bel}(\boldsymbol{Y}_{t-1})$. Étant donné les deux hypothèses que nous avons mentionnées, il est aisé de trouver cette expression pour chaque valeur de $\boldsymbol{Y}_t$:

$$\begin{aligned}
\text{bel}(\boldsymbol{Y}_t = \boldsymbol{y}) &\propto \text{p}(\boldsymbol{X}_t|\boldsymbol{Y}_t = \boldsymbol{y}, \boldsymbol{X}_{1:t-1}, \boldsymbol{U}_{1:t})\,\text{p}(\boldsymbol{Y}_t = \boldsymbol{y}|\boldsymbol{X}_{1:t-1}, \boldsymbol{U}_{1:t}) \\
&\propto \text{p}(\boldsymbol{X}_t|\boldsymbol{Y}_t = \boldsymbol{y}) \int_{\boldsymbol{y}'} \text{p}(\boldsymbol{Y}_t = \boldsymbol{y}|\boldsymbol{Y}_{t-1} = \boldsymbol{y}', \boldsymbol{U}_{1:t})\,\text{p}(\boldsymbol{Y}_{t-1} = \boldsymbol{y}'|\boldsymbol{X}_{1:t-1}, \boldsymbol{U}_{1:t}) \\
&\propto \text{p}(\boldsymbol{X}_t|\boldsymbol{Y}_t = \boldsymbol{y}) \int_{\boldsymbol{y}'} \text{p}(\boldsymbol{Y}_t = \boldsymbol{y}|\boldsymbol{Y}_{t-1} = \boldsymbol{y}', \boldsymbol{U}_t)\,\text{bel}(\boldsymbol{Y}_{t-1} = \boldsymbol{y}') \qquad \text{(I.1)}
\end{aligned}$$

L'application de cette idée à donné naissance à un algorithme général : le filtrage bayésien (voir algorithme I.1). Cet algorithme est composé de deux étapes majeures :

Prédiction du nouvel état : Dans cette étape (voir ligne 2), le robot utilise le contrôle $\boldsymbol{u}_t$ et la fonction de transition pour prédire l'état futur, sans tenir compte de la mesure. Le robot obtient donc la distribution de probabilité avant observation $\text{p}(\boldsymbol{Y}_t|\boldsymbol{X}_{1:t-1}, \boldsymbol{U}_{1:t})$ qu'on note $\overline{\text{bel}}(\boldsymbol{Y}_t)$. Intuitivement, dans cette étape, on déplace les probabilités d'un état à l'autre en fonction des actions du robot. Par exemple, supposons que le robot se trouve devant une porte qu'il croit fermée *i.e.* $\text{bel}(\boldsymbol{Y}_{t-1} = \text{fermé})$ est élevé. Si le robot essaye d'ouvrir la porte, il est naturel de penser que $\overline{\text{bel}}(\boldsymbol{Y}_t = \text{ouvert})$ va augmenter.

Incorporation de la mesure : Dans cette étape (voir ligne 3), la mesure $\boldsymbol{x}_t$ est incorporée pour calculer $\text{bel}(\boldsymbol{Y}_t)$. Les perceptions sont utilisées pour moduler les croyances. Pour continuer notre exemple, si les capteurs du robot indiquent que la porte est toujours fermée (par exemple si la distance mesurée n'a pas augmenté), alors $\text{bel}(\boldsymbol{Y}_t = \text{ouvert})$ va diminuer.

Algorithme I.1: L'algorithme général du filtrage bayésien.

Données : $\text{bel}(\boldsymbol{Y}_{t-1})$, $\boldsymbol{x}_t$, $\boldsymbol{u}_t$
Résultat : $\text{bel}(\boldsymbol{Y}_t)$

1 **pour chaque** $\boldsymbol{y}_t$ **faire**
2 $\quad \overline{\text{bel}}(\boldsymbol{y}_t) = \int_{\boldsymbol{y}_{t-1}} \text{p}(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}, \boldsymbol{u}_t)\,\text{bel}(\boldsymbol{y}_{t-1})$;
3 $\quad \text{bel}(\boldsymbol{y}_t) = \eta \text{p}(\boldsymbol{x}_t|\boldsymbol{y}_t)\,\overline{\text{bel}}(\boldsymbol{y}_t)$;
4 **fin**
5 **retourner** $\text{bel}(\boldsymbol{Y}_t)$

Pour implémenter cette approche, il reste à définir :

– Comment représenter la distribution de probabilité bel $(\boldsymbol{Y}_t)$. Cette question est ardue car l'état est souvent une variable de grande dimension. On peut distinguer les méthodes paramétriques (qui supposent que les densités de probabilités suivent une loi connue, en général une gaussienne ou un mélange de gaussiennes) et les méthodes non-paramétriques du type filtre particulaire ([Thr02]).
– Quel modèle de transition utiliser.
– Quel modèle de capteur utiliser.

Par exemple, le filtre de Kalman [7] correspond au cas où l'on suppose que l'état, les observations et les actions sont représentés par des distributions gaussiennes et que les modèles de transition et de capteurs sont linéaires.

Ces algorithmes peuvent, en principe, être utilisés pour tout type de problème, moyennant une représentation adéquate des croyances et des modèles adaptés. Cependant, leur application la plus courante en robotique concerne les problèmes de localisation et le SLAM.

1.4.3. SLAM

Comme nous l'avons dit, les approches probabilistes sont les plus fructueuses pour résoudre le problème du SLAM. Dans le cadre probabiliste, le problème du SLAM est d'estimer la probabilité *a posteriori* de la position du robot et de la carte.

Localisation Si l'on omet temporairement la construction de la carte pour se concentrer sur la localisation, l'état du robot est défini dans un repère en 2D par sa position (x, y) et son orientation θ. L'état est donc représenté par un vecteur à trois dimensions $\boldsymbol{y} = (x, y, \theta)$ appelé pose.

Dans ce cas, le modèle du monde donne l'évolution de la pose en fonction des actions du robot ; il est donc appelé modèle de déplacement (*motion model*). La plupart des modèles utilisent comme contrôle les informations données par l'odométrie, celles-ci étant en général plus précise que le contrôle basé sur la vitesse. Bien que ces informations soient des mesures, il est possible de les utiliser après coup pour contrôler l'exécution des actions du robot. Le but d'un tel modèle est donc de prédire la probabilité de la pose du robot au temps t, connaissant la pose au temps $t-1$ et le déplacement mesuré. Le mouvement est décomposé en trois étapes (voir Figure I.7), chaque étape étant entachée d'un bruit indépendant : une rotation suivie d'une translation et d'une autre rotation. Si l'on fixe les paramètres des bruits, on peut ainsi calculer la probabilité de $\boldsymbol{y}_t$ en fonction de $\boldsymbol{y}_{t-1}$ et $\boldsymbol{u}_t$.

Le modèle des capteurs donne la probabilité d'une mesure en fonction de la pose : P $(\boldsymbol{x}|\boldsymbol{y})$. Pour la tâche de localisation, il est utile de disposer d'un modèle métrique du capteur, c'est-à-dire un modèle qui donne des informations sur la géométrie du monde. Un tel modèle

7. Ce modèle a été découvert de nombreuses fois de manière (plus ou moins) indépendante. Il a été popularisé par son utilisation dans le programme lunaire américain.

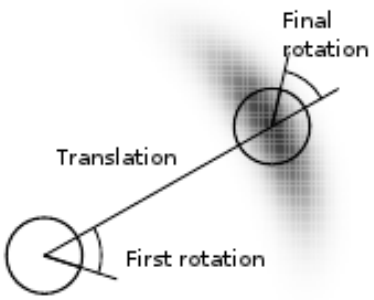


FIGURE I.7.: Modèle de déplacement basé sur l'odométrie. Les zones grisées indiquent la densité de probabilité des positions prévues. Adapté de [TBF05].

permet de connaître à quelle position dans le repère global correspond une donnée du capteur. On peut ainsi reconnaître les objets indépendamment de la position dans laquelle ils ont été observés. Par exemple, si le télémètre repère un obstacle à une distance de 2 m sur l'axe x, connaissant la pose, on peut facilement connaître la position de l'obstacle. À l'opposé, certains capteurs (par exemple un capteur d'odeur) ne permettent pas l'utilisation d'un modèle métrique. Dans ce cas, les perceptions ne permettent que de reconnaître un lieu déjà visité. La plupart des capteurs utilisés en robotique (télémètres, caméras) disposent d'un modèle métrique (voir [TBF05, chap. 6]).

Localisation et cartographie simultanée Dans le cas du problème complet, le but est d'estimer simultanément la pose du robot $\boldsymbol{y}$ et la carte $m = \{m_1, m_2, \ldots, m_N\}$ (voir Section I.1.2.3). On distingue deux versions du problème :

1. L'estimation en ligne, qui consiste à estimer la probabilité de la localisation courante et de la carte : $\mathrm{p}\left(\boldsymbol{y}_t, m | \boldsymbol{x}_{1:t}, \boldsymbol{u}_{1:t}\right)$.
2. L'estimation complète de toute la trajectoire et de la carte : $\mathrm{p}\left(\boldsymbol{y}_{1:t}, m | \boldsymbol{x}_{1:t}, \boldsymbol{u}_{1:t}\right)$.

Le problème comprend deux parties distinctes :

1. l'estimation de la position du robot et la position de chaque objet de la carte qui est une variable continue (de très grande dimension).
2. l'estimation des correspondances entre les entrées sensorielles et les objets (ce problème est connu sous le nom de *correspondence problem*) qui est un problème discret. Le problème consiste à reconnaître à quel objet de la carte une entrée sensorielle $\boldsymbol{x}$ correspond. Pour les cartes basées sur les points de repère, cela implique de reconnaître le point de repère à partir de $\boldsymbol{x}$.

Parmi les premiers travaux utilisant cette approche pour le problème de la navigation, citons [ME85] qui apprend des grilles d'occupation (voir Section I.1.2.3). D'autres types de

cartes métriques ont été utilisés représentant la forme locale de l'environnement à l'aide de primitives géométriques (voir [CL85 ; LDC92 ; HBC96 ; Cas+99 ; TBF00]). Certaines approches utilisent une décomposition irrégulière de l'environnement, par exemple sous forme de d'arbre de densité (voir [Thr98b]). Parmi les approches utilisant une carte topologique, citons [KW95 ; DMS99 ; Ang+08].

Les techniques actuelles sont robustes et applicables à de grands environnements (voir par exemple [Mon03]). Ces techniques ont été utilisées lors du concours *Grand Challenge* organisé par le DARPA [8].

Les techniques d'apprentissage peuvent être utilisées pour construire les modèles de capteurs et estimer le problème de correspondance.

1.5. Conclusion

Parmi les différentes approches en robotique, les algorithmes probabilistes sont reconnus pour leur robustesse. Bien que nous ne fassions pas de cartographie, le problème de reconnaissance de lieux sémantiques peut bénéficier de ces approches. En retour, la reconnaissance de lieux sémantiques peut fournir un complément idéal aux techniques de navigation pour certains problèmes difficiles à résoudre, tels le problème du robot kidnappé et le problème de la fermeture de boucle (voir section III.2).

Nous avons rencontré la problématique de la reconnaissance à deux reprises dans cette partie : pour les stratégies basées sur les réflexes (voir Section I.1.2.2, stratégie 3) et pour le problème de correspondance.

Reconnaître des lieux d'apparence similaire nécessite d'avoir accès à beaucoup de détails. Or la plupart des approches du SLAM se basent sur des télémètres lasers qui fournissent une information robuste mais assez fruste. La vision au contraire fournit des informations riches et semble donc adaptée à la reconnaissance de lieux. Les travaux en SLAM utilisent de plus en plus les techniques de vision par ordinateur. Cependant, il y a deux écueils à éviter. Si on prend en compte trop de détails, le système va devenir sensible à de petites modifications qui risquent de rendre impossible la localisation (problème de la variabilité perceptuelle). On utilise alors des techniques de traitement du signal pour extraire des informations dans les mesures et réduire la variabilité de l'apparence des lieux. *A contrario*, si on réduit trop la variabilité, deux lieux différents ne sont plus distinguables : c'est le problème du *perceptual aliasing* (qu'on pourrait traduire par « confusion perceptuelle »).

Dans la suite de notre travail, nous utiliserons la vision comme modalité. Nous abordons la problématique de la représentation des images dans la section suivante, consacrée au domaine de la vision par ordinateur. Les problématiques de la reconnaissance visuelle seront abordées plus en détail dans le chapitre II.

8. Voir http://en.wikipedia.org/wiki/DARPA_Grand_Challenge pour une description du concours.

2. Vision par ordinateur

Lorsqu'on présente une image à un être humain, il l'interprète spontanément en terme d'objets, de texte, de personnes, voire d'intentions. Ce processus est en général inconscient et très rapide (quelques centaines de millisecondes).

Informatiquement, une image est représentée par une matrice de pixels, chaque pixel étant codé par un ou plusieurs nombres. Par exemple, la Figure I.8 montre une image et sa représentation informatique. Là où nous interprétons spontanément le contenu sémantique de l'image, l'ordinateur n'a accès qu'à la valeur des pixels [9]. Cette différence entre l'interprétation d'une image et l'information contenue dans la valeur des pixels est appelée « gouffre sémantique ».

L'image informatique est obtenue à partir d'un processus complexe de projection de la scène 3D (par l'appareil optique), de quantification spatiale (pixel) et de quantification de l'intensité (niveau discret de luminance). On appelle pré-image la scène 3D telle qu'elle est réellement.

La vision par ordinateur vise à extraire de l'information d'une ou plusieurs images afin de résoudre une tâche. Les tâches classiques de vision par ordinateur sont :

- la segmentation d'image et l'extraction de contours ;
- la reconnaissance d'objets, de scènes et de visages ;
- plus récemment, l'indexation de larges bases de données d'images, appelé *Content Based Image Retrieval (CBIR)* en anglais, en vue de retrouver des images similaires.

La question fondamentale de la vision par ordinateur, qui distingue cette discipline du domaine voisin du traitement d'image, peut être vue comme un problème inverse : comment reconstituer la scène en 3D à partir de l'image en 2D ? De ce point de vue, la plupart des problèmes de vision sont des problèmes mal-posés. C'est pourquoi la plupart ne peuvent pas être résolus dans une approche purement ascendante (*data-driven* ou *bottom-up*). Ce problème est particulièrement important dans les applications de la vision par ordinateur pour un robot, celui-ci étant intrinsèquement situé dans la scène 3D. La difficulté principale est qu'une légère modification des conditions de prise de vue (angle de prise de vue, luminance, zoom) génère une image très différente d'un point de vue informatique.

L'importance des connaissances *a priori* peut être illustrée à l'aide de la Figure I.9. Cette figure est interprétée de façon stable par les individus : la première et la troisième ligne sont perçues convexes (à l'exception du 5^{e} hémisphère de la 3^{e} ligne), tandis que les deux autres sont perçues concaves. Cependant, cette figure est intrinsèquement ambiguë : si l'on retourne la feuille, une ligne interprétée comme concave est maintenant convexe. L'interprétation de

9. Un exemple des succès et des limites de la vision par ordinateur est l'utilisation des images dans les systèmes de type CAPTCHA destinés à la lutte contre les robots sur internet (voir http://en.wikipedia.org/wiki/CAPTCHA) : le principe consiste à demander à l'utilisateur de reproduire des lettres déformées sur des images bruitées. En se concentrant sur un site particulier, les meilleurs programmes atteignent des taux de reconnaissance de l'ordre de 20% à 30%.

(a)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	0
0	255	100	100	100	255	150	150	255	150	150	255	200	200	200	200	255	0
0	255	255	100	255	255	150	150	150	150	150	255	200	255	255	255	255	0
0	255	255	100	255	255	150	255	150	255	150	255	200	255	200	200	255	0
0	255	255	100	255	255	150	255	255	255	150	255	200	255	255	200	255	0
0	255	100	100	100	255	150	255	255	255	150	255	200	200	200	200	255	0
0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)

FIGURE I.8.: Une image et sa représentation informatique. Extrait de [Tir10].

cette figure repose sur la connaissance acquise, lors de nos expériences quotidiennes, que les sources lumineuses sont situées en hauteur et donc la partie basse d'un objet convexe est sombre. Ainsi, les parties dont la partie basse est plus sombre sont interprétées comme convexes et celles dont la base est plus claire sont interprétées comme concaves.

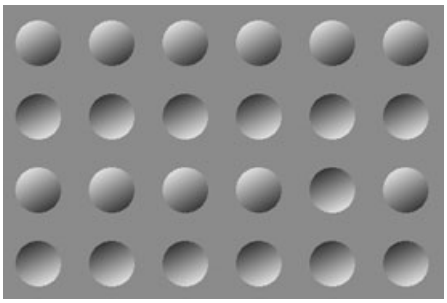


FIGURE I.9.: Exemple de figure ambiguë.

Fondamentalement, la perception est donc un mécanisme d'inférence (cette idée a été exprimée pour la première fois par HELMOLTZ [Hel67]). Le formalisme le plus adapté pour l'utilisation de connaissances *a priori* est le formalisme bayésien (voir KERSTEN, MAMASSIAN et YUILLE [KMY04]). Ce formalisme permet de combiner les connaissances antérieures avec les observations pour inférer une interprétation de la scène. Il permet aussi (comme nous l'avons vu dans la section I.1.4 sur la robotique probabiliste) de mettre à jour les hypothèses à partir de nouvelles observations.

Nous nous concentrons dans cette partie sur les techniques permettant de décrire une image. Les problématiques de la reconnaissance visuelle seront discutées au chapitre II.

Bien que la vision par ordinateur soit un domaine différent, les techniques du traitement d'image y sont abondamment employées et forment souvent une étape cruciale des algorithmes. On distingue trois grandes classes d'opérations en traitement d'image :

1. les opérations ponctuelles : changement de contraste, modification de couleurs, modification de l'histogramme. Dans ces opérations, la valeur de chaque pixel est modifiée indépendamment des autres pixels.
2. les opérations de voisinage : filtrage (linéaire ou non), morphologie mathématique, *etc.* Dans ces opérations, la nouvelle valeur d'un pixel est calculée à partir d'un voisinage autour de ce pixel.
3. les changements d'espace de représentation : transformée de Fourier, transformée en ondelettes, *etc.* Ici, l'image est transformée globalement en une nouvelle représentation ce qui permet de mettre en lumière d'autres propriétés de l'image. La transformée de Fourier permet d'analyser le contenu fréquentiel des images c'est-à-dire à quelle fréquence la valeur des pixels varie. La transformée en ondelettes et l'analyse multirésolution

(voir [Mal99]) permettent d'analyser en même temps les propriétés spatiales et spectrales d'une image.

Nous ne pouvons détailler toutes les techniques que nous rencontrerons. Nous renvoyons le lecteur intéressé vers les ouvrages spécialisés (par exemple [Pra07 ; Sze10]).

Une source d'inspiration importante en vision par ordinateur est évidemment la vision humaine. Ainsi, nous ferons parfois référence à certaines de ses caractéristiques. Là encore, le lecteur est invité à se reporter aux ouvrages de référence (par exemple [Zek93]).

Comme nous l'avons dit, la vision par ordinateur doit éviter le double écueil de la variabilité perceptuelle et de la confusion perceptuelle. Nous présentons dans un premier temps (Section I.2.1) les caractéristiques utilisées habituellement en vison par ordinateur. Ces caractéristiques sont, en général, exprimées sous forme d'un vecteur appelé « descripteur » ou encore « signature ». On peut distinguer deux grands types de descripteurs que nous présenterons successivement :

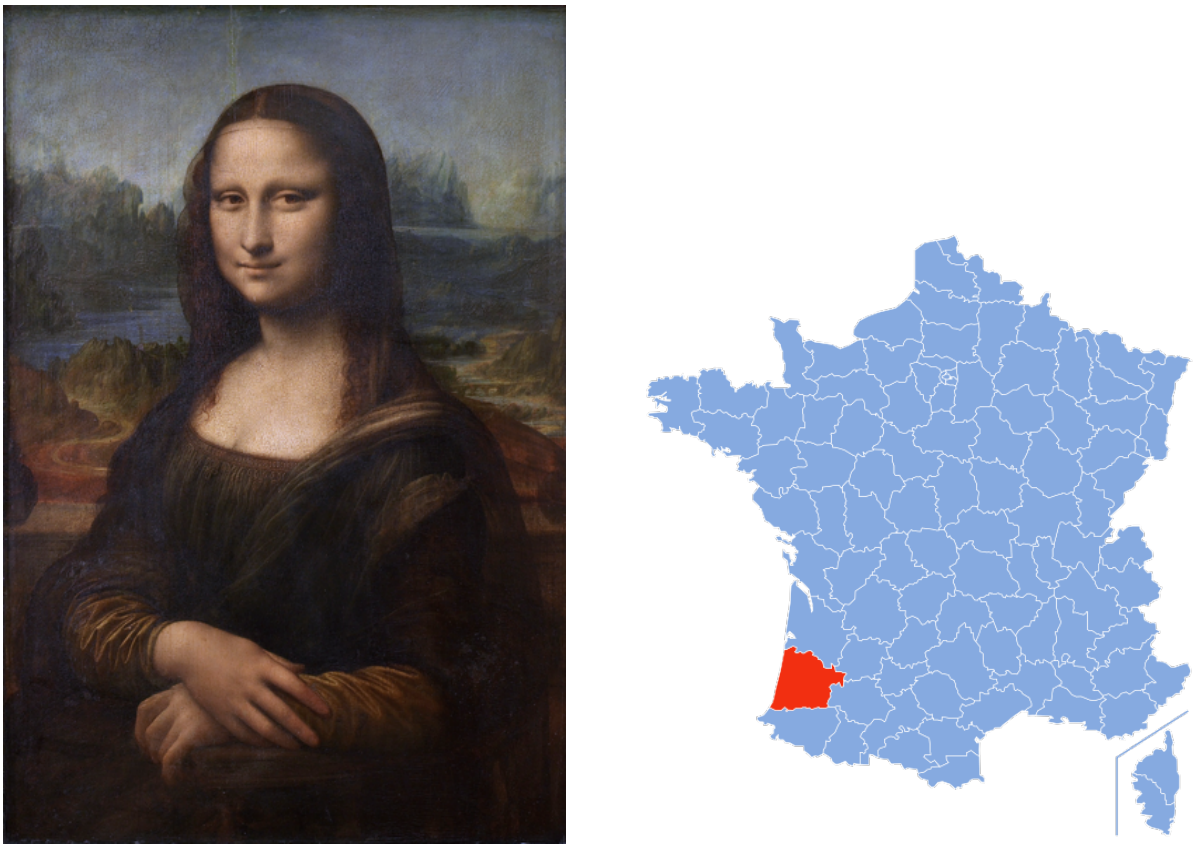
- Les descripteurs globaux, c'est-à-dire les descripteurs dont la valeur dépend de tous les pixels de l'image (voir Section I.2.2).
- Les descripteurs locaux, dont la valeur est calculée à partir d'un sous-ensemble des pixels de l'image. Nous les présentons dans la Section I.2.4 après avoir présenté les techniques d'extraction de régions d'intérêt (Section I.2.3).

2.1. Description phénoménologique des images naturelles

Les images naturelles sont définies par opposition aux images fabriquées ou manipulées par l'homme (images de synthèse, tableaux, dessins, *etc.*). La figure I.10 présente des exemples d'images naturelles et d'images non-naturelles. Le chapitre II se concentrera sur une sous-classe des images naturelles : les scènes. L'étude des images naturelles présente un grand intérêt parce que, bien que représentant une fraction infime de l'ensemble des images possibles, elles forment une part non-négligeable de l'ensemble des images que nous rencontrons quotidiennement [10]. C'est à ce type d'images que sera confronté notre système robotique, aussi, nous nous intéresserons uniquement à celles-ci.

Nous présentons les grandeurs perceptuelles utilisées à la fois par les humains et par les algorithmes de vision par ordinateur pour accéder au contenu des images.

10. On notera d'ailleurs que la plupart des études sur le système visuel humain utilisent des images non-naturelles comme les barres, les réseaux (*gratings*), *etc.*



(a)

(b)

FIGURE I.10.: Exemple d'images artificielles (a) et naturelles (b).

2.1.1. Intensité

La valeur des pixels code en général l'intensité lumineuse locale d'une zone de la pré-image [11]. C'est l'information de base utilisée par les algorithmes de vision par ordinateur. La valeur enregistrée par un pixel dépend de l'imageur, de ses réglages, de la luminosité ambiante, *etc.* Ainsi, la valeur brute en niveau de gris est rarement utilisée telle quelle, la plupart des algorithmes utilisant des gradients de niveau de gris.

2.1.2. Couleur

Il existe différents moyens de représenter la couleur dans une image informatique. Le codage le plus courant est le codage RGB où chaque pixel est représenté par 3 valeurs indépendantes, chacune représentant un canal (une couleur est donc représentée par un point dans un espace euclidien à 3 dimensions). La couleur finale est obtenue par synthèse additive. L'inconvénient majeur de ce modèle de couleur est qu'il dépend de l'appareil utilisé pour représenter l'image.

D'autres descriptions plus proches de la perception humaine des couleurs existent (appelées espaces perceptifs). L'idée principale est de séparer les informations de couleur et de luminance. C'est d'ailleurs ainsi que procède le cortex visuel (voir [SLC90]). On peut citer ([Pra07]) :

- L'espace HSV (*Hue, Saturation, Value*) qui code les couleurs selon un système de coordonnées cylindriques : l'angle autour de l'axe central est la teinte (*hue*), qui représente la sensation de couleur, indépendamment de la luminosité ; la distance à l'axe correspond à la saturation qui représente le degré de couleur (pâle ou foncé) ; enfin la distance le long de l'axe principal correspond à la brillance (*value* ou *brightness*). La séparation entre la couleur et la teinte est intéressante mais l'espace HSV a été critiqué car les 3 composantes sont perceptivement corrélées.
- L'espace Lab : L représente la luminosité, a et b représentent la position de la couleur sur des axes d'opposition de couleurs (rouge/magenta *versus* vert pour a, bleu *versus* jaune pour b). Ce système de mesure est très proche de la perception humaine des couleurs.

Le passage du codage RGB à l'un de ces codages s'effectue indépendamment pour chaque pixel, sans tenir compte du voisinage.

La notion de couleur est liée à la longueur d'onde des radiations lumineuses reçues. Cependant, la couleur ne se réduit pas à la longueur d'onde. On sait par exemple que deux sources ayant des caractéristiques spectrales différentes peuvent être perçues comme étant de la même couleur (ce phénomène est appelé métamérisme ; voir [Pra07, p. 25]). À l'opposé, d'autres expériences reportent que nous percevons un objet de la même couleur à différents moments de la journée alors que les propriétés spectrales de la source et donc les longueurs d'ondes réfléchies changent. Certains algorithmes ont cherché à reproduire ce phénomène

11. Il existe des images pour lesquelles la valeur du pixel code pour tout à fait autre chose. Par exemple, dans une image échographique, la valeur d'un pixel code l'intensité du signal sonore réfléchi par les tissus et donc leur densité.

appelé « constance des couleurs », sans y arriver totalement, le plus connu étant Retinex (voir [MPS10]) : dans ces algorithmes, la notion de couleur est définie en fonction des pixels voisins.

2.1.3. Texture

Une texture est une zone de l'image qui a des caractéristiques homogènes, ces caractéristiques ne se réduisant pas uniquement à la couleur ou à la luminance, par exemple, un motif basique qui se répète ou des caractéristiques fréquentielles. La notion de texture est difficile à définir formellement ; il y a plusieurs approches de cette notion.

La description perceptuelle (voir [TMY78]) cherche à exprimer les propriétés des textures de la même manière que le ferait un humain. Ces descripteurs sont donc basés sur des expériences psycho-physiques qui ont permis de définir trois caractéristiques fortement corrélées avec la manière utilisée par les sujets pour classer des textures et les algorithmes pour les calculer :

1. Le contraste qui dépend de la dynamique des niveaux de gris, de la netteté des contours et de la période des motifs répétés.
2. La directionnalité (*directionality*), c'est-à-dire la présence d'une orientation dominante (quelle qu'elle soit) dans la texture.
3. La rugosité (*coarseness*) c'est-à-dire la taille des éléments de texture.

L'approche statistique (voir [HSD73]) se base sur la matrice de co-occurrence des valeurs de niveau de gris, c'est-à-dire une approximation de la statistique d'ordre 2. Cette matrice de co-occurrence est collectée dans un voisinage autour de chaque pixel. Différentes valeurs statistiques (que nous décrivons en Appendice C) sont ensuite calculées pour exprimer les propriétés de la texture pour chaque pixel.

On peut aussi utiliser la sortie d'un banc de filtres orientés et multi-échelles pour caractériser les textures (voir [Des03]). On obtient alors plusieurs valeurs pour chaque pixel. Il existe plusieurs techniques pour réduire la quantité d'information :

1. calculer la moyenne (énergie) et l'écart-type de chaque filtre, ce qui offre une description compacte ;
2. calculer la sortie uniquement pour certains points d'intérêt.

Les filtres de Gabor (voir plus bas) ont été les plus utilisés dans ce contexte.

2.1.4. Contours

Les contours (*edges*) sont les régions de l'image où l'intensité varie rapidement. Ces régions se distinguent du bruit (qui peut aussi causer des variations rapides de l'intensité) par le fait qu'elles possèdent une structure. Mathématiquement, un contour est défini comme une discontinuité dans la valeur des pixels. Un contour est caractérisé par le gradient local (l'amplitude de la variation) et son orientation (la direction de variation).

Les discontinuités sur l'image sont en général provoquées par :
- des discontinuités de profondeur ;
- des discontinuités d'orientation des surfaces ;
- des changements des propriétés des matériaux ;
- des variations de l'illumination de la scène.

Ainsi, dans les cas simples, les contours donnent directement la frontière des objets (ou de leurs parties) ou les occlusions entre objets.

L'idée de s'intéresser aux contours est motivée par deux approches. Premièrement, l'étude des mécanismes neurophysiologiques de la perception : les cellules simples du cortex visuel sont sensibles aux contours orientés localisés dans une partie de l'espace visuel, appelée champ récepteur (voir [HW62]) ; ces cellules sont souvent décrites comme des détecteurs de contours. On sait depuis les travaux de Daugman [Dau80] ; Jones et Palmer [JP87] que la réponse des cellules simples du cortex visuel primaire ressemble beaucoup à la partie réelle du filtre de Gabor (voir plus bas). D'autres modèles ont été proposés : le modèle log-Gabor ([Fie87 ; KWG94]) ou log-normal ([Hér10]).

Deuxièmement, les approches théoriques du codage des images (voir [Fie87 ; Fie94 ; OF00]) ont aussi montré l'importance des contours. Ces études cherchent à définir un moyen de codage optimal, sous certains critères, des images (voir Willmore, Watters et Tolhurst [WWT00]) : on cherche donc à remplacer le codage matriciel des pixels par une autre représentation qui utilise des composants plus abstraits. Un exemple de code est la transformée de Fourier dans laquelle les images sont décrites à l'aide de fonctions trigonométriques à toutes les fréquences. Suivant le domaine, les composants sont appelées « axes », « filtres », « atomes » ou « cellules ».

Un code efficace est un code peu redondant : en effet, si les réponses des filtres sont corrélées, certaines unités codent la même information [12]. Le but d'un code efficace est donc de trouver un code dont les axes sont statistiquement indépendants ([Att54 ; Bar61]). Un code réalisant cette condition est appelé « code factoriel ». De manière, générale, la recherche du code le plus indépendant possible implique, au minimum, de décorréler les réponses des filtres *i.e.* d'obtenir une matrice de covariance diagonale (cette opération est appelée blanchiment. Cependant, sans autre contrainte, il n'y a pas de solution unique au problème.

Si l'on fait l'hypothèse que les nouveaux axes doivent être orthogonaux, la recherche d'une solution au problème de décorrélation aboutit à l'Analyse en Composantes Principales (ACP), qui est une technique de décorrélation permettant d'obtenir une nouvelle base orthonormale dans laquelle la matrice de covariance est diagonale. Les nouveaux axes de description sont donc décorrélés. Si la répartition des niveaux de gris des images était gaussienne, l'ACP fournirait un codage optimal : en effet, les moments d'ordre supérieur à 2 sont nuls pour une gaussienne (*i.e.* toute la redondance est exprimée dans la matrice de covariance) donc éliminer la corrélation entre les composants élimine toute la redondance. De plus, les axes

12. Notons que la redondance peut être utile pour faire face au cas d'unités défaillantes (voir [OF97]).

sont ordonnés en fonction de la part de variance qu'ils restituent : le premier axe représente le maximum de variance, le deuxième axe représente le maximum de variance restante, *etc.* (en terme de cellules, la première cellule est activée plus souvent que la seconde, *etc.*). Ce principe permet de compresser une image en ne considérant que les premières composantes principales.

Cependant, on sait que la distribution des niveaux de gris dans l'image est non-gaussienne ([Fie94]) : les images naturelles présentent un kurtosis normalisé positif. Rappelons que le kurtosis normalisé [13] est défini par :

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3 \tag{I.2}$$

où μ_4 est le moment centré d'ordre 4 et σ l'écart type (la constante de normalisation est introduite de sorte que $\gamma_2 = 0$ pour une distribution gaussienne). Le kurtosis normalisé mesure à la fois le caractère piqué et l'aplatissement d'une courbe par rapport à une gaussienne (*i.e.* le fait que les queues des distributions sont larges). Les distributions plus aplaties que les gaussiennes ont un kurtosis positif (voir Figure I.11). Dans ces distributions, la variance est surtout due à des variations importantes, loin de la moyenne, mais peu fréquentes, ce qui est caractéristique des contours dans les images naturelles.

Or, l'ACP produit des filtres non-localisés ([Fie99]). Pour chercher à expliquer le caractère localisé des filtres corticaux en présence de données non-gaussiennes, il est nécessaire de chercher des solutions non-orthogonales au problème de décorrélation. Enlever cette hypothèse conduit à l'Analyse en Composantes Indépendantes (ACI) (voir Comon [Com94] ; Hyvärinen et Oja [HO00]). De manière générale, le problème de l'ACI n'a pas toujours de solution. De plus, d'un point de vue algorithmique, la recherche de la solution passe par la minimisation d'un critère : par exemple minimiser l'information mutuelle entre les composantes, ce qui conduit au principe infomax (voir [Lin88 ; BS95]), ou maximiser la non-gaussianité, ce qui conduit à l'algorithme fastICA ([Hyv99]). Ces algorithmes font donc des hypothèses supplémentaires, ce qui peut conduire à ne pas trouver la solution quand elle existe.

Olshausen et Field [OF96] ; Bell et Sejnowski [BS97]) ont appliqué des algorithmes d'ACI à des images naturelles. Leurs résultats (voir Figure I.12) ont montré que les composantes indépendantes des images naturelles sont des filtres orientés et localisés, proches des ondelettes. Ces résultats ont permis d'expliquer à l'aide de la théorie de l'information pourquoi les contours sont intéressants (et pourquoi les cellules simples du cortex visuel primaire se comportent ainsi). Ils fournissent aussi des fondements théoriques solides sur l'utilité de la décomposition en ondelettes. Fondamentalement, l'importance des contours est donc liée à la statistique des images naturelles. De manière générale, on peut dire que les contours contiennent l'information de l'image.

13. Le mot est forgé à partir du grec *κυρτός* qui signifie « courber »

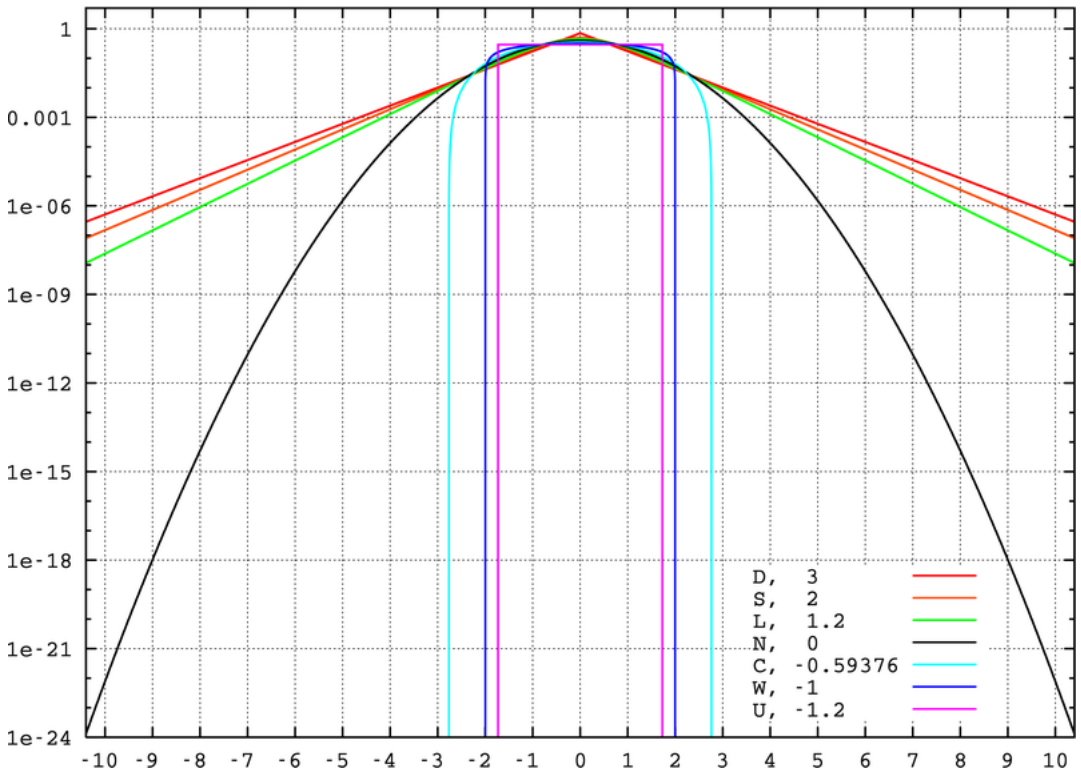


FIGURE I.11.: Exemple de distributions et de leur kurtosis sur une échelle logarithmique :
D : distribution de Laplace (courbe rouge, $\gamma_2 = 3$)
S : distribution hyperbolique (courbe orange, $\gamma_2 = 2$)
L : distribution logistique (courbe verte, $\gamma_2 = 1{,}2$)
N : distribution normale (courbe noire, $\gamma_2 = 0$)
C : distribution du cosinus (courbe cyan, $\gamma_2 = -0{,}593\ 762$)
W : distribution de Wigner (courbe bleue, $\gamma_2 = -1$)
U : distribution uniforme (courbe magenta curve $\gamma_2 = -1{,}2$).
Source : http://en.wikipedia.org/wiki/Kurtosis.

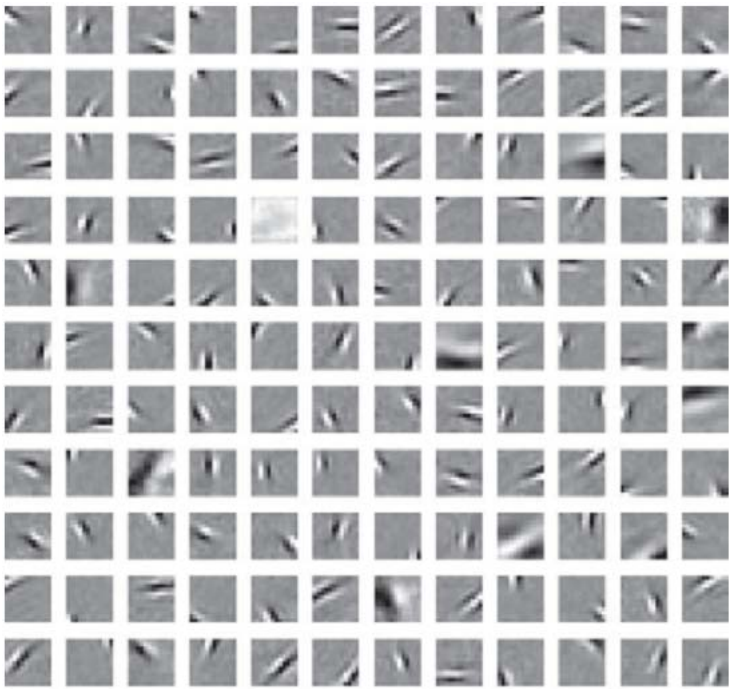
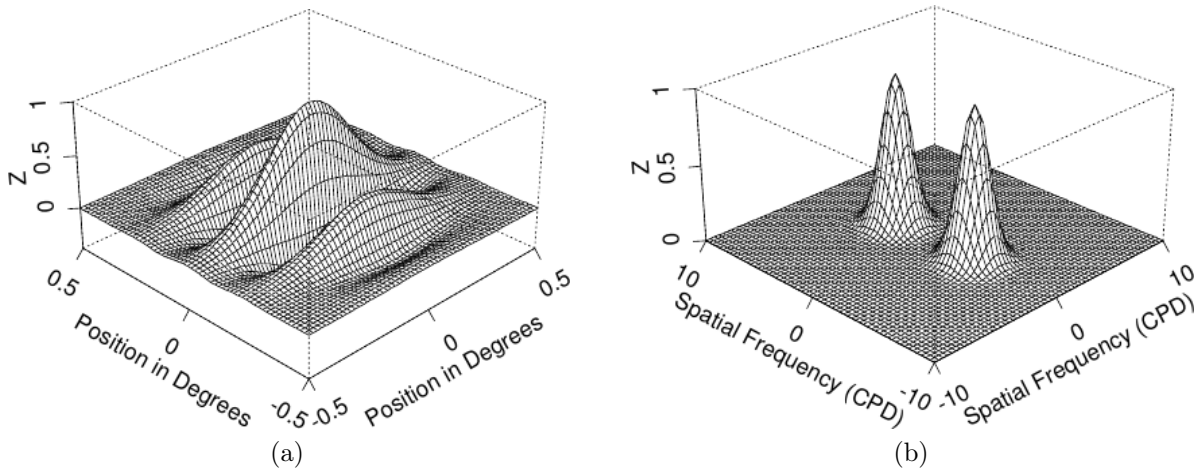


FIGURE I.12.: Base de fonctions optimale déterminée par [OF96] sur des patchs de taille 12×12 tirés d'images naturelles. Ces fonctions, déterminées empiriquement, sont très similaires à des filtres passe-bande, orientés et localisés.



(a) (b)

FIGURE I.13.: (a) partie réelle du filtre de Gabor 2D (b) transformée de Fourier du filtre de Gabor. Tiré de [Dau09].

Dans les images naturelles, les contours ne sont, en général, pas faciles à détecter à cause de la présence d'ombres, de flou... On rencontre souvent dans les images non-triviales, des contours fragmentés, des contours non-détectés ou au contraire des faux positifs. De plus, les contours existent à plusieurs échelles. Les théories de l'espace-échelle (voir [PM90 ; Lin96]) fournissent un cadre rigoureux (lié à l'analyse multirésolution) pour détecter les contours à plusieurs échelles.

Enfin, au-delà des discontinuités de niveau de gris, on peut s'intéresser à des contours dans des domaines plus abstraits (texture, profondeur).

D'un point de vue pratique, le détecteur de contour le plus utilisé est sans doute le filtre de Gabor. Ce filtre est donné par le produit d'une sinusoïde complexe et d'une enveloppe gaussienne (voir Figure I.13) :

$$\text{Gabor}\,(x,y) = \frac{1}{2\pi\sigma_x^2\sigma_y^2}\exp\left(\frac{x_r^2}{\sigma_x^2}+\frac{y_r^2}{\sigma_y^2}\right)\times\exp\left(-\imath\left[\nu_x(x-x_0)+\nu_y(y-y_0)\right]\right) \tag{I.3}$$

Avec :

$$\begin{aligned} x_r &= \quad (x-x_0)\cos(\theta)+(y-y_0)\sin(\theta)\\ y_r &= -(x-x_0)\sin(\theta)+(y-y_0)\cos(\theta)\end{aligned}$$

Où (x_0, y_0) est le centre du champ récepteur, $\left(\sigma_x^2, \sigma_y^2\right)$ spécifient la largeur du support spatial du filtre, θ son orientation et (ν_x, ν_y) sont les fréquences spatiales dans les directions x et y respectivement.

D'un point de vue historique, le filtre de Gabor entretient une relation étroite avec la théorie des ondelettes [14].

2.2. Caractérisation globale des images

Les descripteurs globaux utilisent tous les pixels d'une image pour calculer une signature. Le vecteur obtenu est donc sensible aux objets présents dans l'image, à leurs positions, à l'illumination globale, *etc.*

Bien qu'utilisant tous les pixels d'une image, il est possible d'inclure dans ces descripteurs une information de nature géométrique en découpant l'image en sous-images. La description globale peut être obtenue en concaténant la description de chaque sous-image. On parle alors de géométrie grossière.

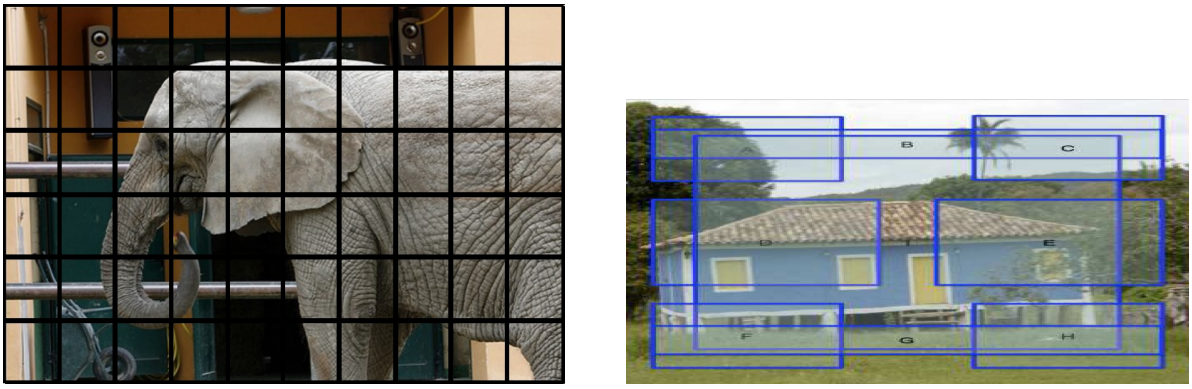
Le découpage le plus utilisé est sans doute le découpage selon une grille régulière [15], comme sur la figure I.14a (en général, chaque bloc fait moins de 100 pixels de côté). Ce type de découpage est utilisé dans les descripteurs GIST et CENTRIST que nous présentons en détail dans la suite de cette section. L'avantage d'une segmentation régulière de l'image est de proposer une représentation finale, par exemple un vecteur de signatures locales, qui encode indirectement une information sur l'organisation spatiale de la scène. L'inconvénient d'un point de vue informationnel est que l'échantillonnage de la scène étant arbitraire, certaines régions sont redondantes tandis que d'autres n'ont aucun contenu intéressant.

On peut aussi citer le découpage en zones exploitant la géométrie des images (voir I.14b) : dans ce cas l'image est divisée en régions en fonction de zones que l'on peut identifier généralement dans les images, par exemple le haut, le centre et le bas de l'image. L'objectif est d'utiliser des informations *a priori* sur la prise de vue des images : on considère ainsi que le zone centrale de l'image contient l'essentiel de l'information ou encore que le ciel a de très fortes chances de se trouver en haut de l'image. Ce découpage est intéressant pour les images prises par un observateur humain (voir [TO03]) mais comme nous le verrons au chapitre III, cette hypothèse n'est pas toujours vérifiée pour les images prises par un robot.

Les descripteurs globaux présentent l'avantage d'être rapides à calculer et d'offrir une représentation compacte des images (en général, ce sont des vecteurs de quelques centaines de dimensions). De plus, la taille de la description ne dépend, en général, pas de la taille des images ou de leur contenu, ce qui est intéressant pour les algorithmes de classification. En revanche, ils décrivent les images de manière très rudimentaire, ce qui ne permet pas toujours de les différencier (en particulier, toutes les informations locales sur les images sont perdues). Leur utilisation est cependant justifiée par des études en psychologie qui montrent

14. Gabor, pressentant leur importance, avait nommé ses filtres des *logons*, néologisme formé du grec *λόγος*, « parole, discours, information » (voir [Dau09]).

15. Les appareils photos numériques récents superposent d'ailleurs une telle grille lors du cadrage de la photo.



(a) (b)

FIGURE I.14.: Découpages de l'image permettant d'incorporer des informations géométriques dans une description globale : (a) découpage en grille régulière (b) découpage en zones. Tiré de [Tir10].

que l'aspect général d'une image est perçu de manière assez grossière, indépendamment des nombreux détails qui peuvent y apparaître. Ainsi, le fait qu'une image soit particulièrement floue empêche d'en percevoir les détails mais permet néanmoins d'en comprendre la structure globale (nous reviendrons sur ce point dans la chapitre II).

Il existe de nombreux descripteurs globaux basés sur les informations décrites dans la Section I.2.1. Nous ne ferons que les mentionner ici (voir [Tir10, chap. 1] pour plus de détails). Parmi les descripteurs basés sur les couleurs, citons (voir [MZ98]) : les moyennes et les moments de la distribution, les histogrammes de couleurs, le *color coherence vector* qui est un histogramme double dans lequel on sépare les pixels appartenant à des zones uniformes et les pixels appartenant à des zones non-uniformes. Les descripteurs de texture les plus utilisés sont les bancs de filtres de Gabor (voir [HR04]). Concernant les descripteurs de contours, on peut utiliser des histogrammes de contours.

Nous présentons en détail deux descripteurs basés sur les contours qui ont été utilisés dans la tâche de reconnaissance de lieux.

2.2.1. GIST

Le descripteur GIST (voir page 81 pour une explication du nom) est issu d'une série de travaux à la fois psychologiques et informatiques sur la classification de scènes. Il a été proposé par OLIVA et TORRALBA [OT01] et décrit plus précisément dans TORRALBA et al. [Tor+03] [16]. DENQUIVE et TARROUX [DT02] ont proposé un codage proche.

16. Il y a plusieurs présentations différentes du descripteur. Nous nous basons ici sur une étude du code source disponible sur la page d'Antonio Torralba (http://people.csail.mit.edu/torralba/code/spatialenvelope/) et sur le code tiré de [Dou+09] (http://lear.inrialpes.fr/software).

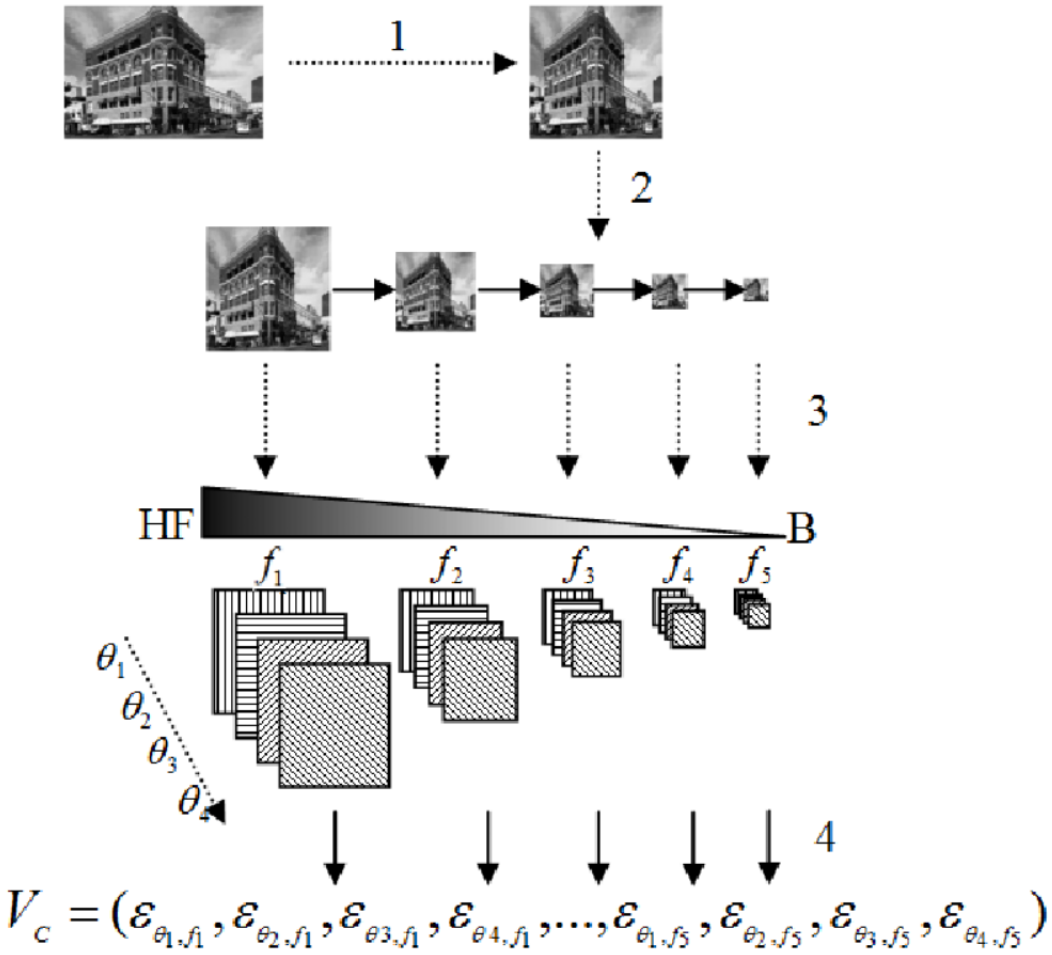


FIGURE I.15.: Descripteur GIST.

Dans un premier temps, les images sont réduites en imagettes carrées, d'une taille comprise entre 32×32 pixels et 128×128 pixels, quelles que soient leur proportions. Les images réduites sont filtrées dans un banc de filtres de Gabor avec n_θ orientations et n_σ échelles (voir Figure I.15) [17]. Les images réduites sont ensuite divisées en une grille régulière de $M \times M$ régions sur lesquelles on calcule la moyenne. Les descripteurs des différentes régions sont ensuite concaténés pour obtenir un vecteur décrivant l'image dans sa globalité. Pour une image couleur, ces opérations sont répétées pour chaque canal.

On obtient ainsi un vecteur de taille $M \times M \times n_\theta \times n_\sigma$ ($\times 3$ pour les images couleur), indépendante de la taille de l'image. La plupart du temps, on utilise $n_\sigma = 4$ échelles et $n_\theta = 6$

17. Les images sont préalablement blanchies.

orientations, ce qui donne 384 dimensions.

Finalement, on utilise une ACP pour réduire la dimension des données (en général, on utilise $D = 80$ composantes). GIST encode donc les composantes principales des orientations moyennes locales à plusieurs échelles.

2.2.2. CENTRIST

Le descripteur *CENsus TRansform hISTogram (CENTRIST)* a été décrit par Wu, Christensen et Rehg [WCR09] pour la tâche de reconnaissance de lieux (voir figure I.16a et [Wu09]). La première étape du descripteur consiste à extraire les contours à l'aide d'un filtre de Sobel ([Sze10, p. 116]).

La deuxième étape est basée sur un encodage de la structure locale du gradient autour de chaque pixel appelée Census Transform (CT) (voir figure I.16b). La valeur de chaque pixel est comparée à celle de ses 8 voisins, ce qui permet de binariser localement l'image : si le pixel central est supérieur ou égal à son voisin, la position correspondante est mise à 1 (sinon à elle est mise à 0). Cette image binaire locale est convertie en un nombre binaire sur 8 bits en parcourant la matrice ligne par ligne. La nouvelle valeur du pixel est donnée par la conversion en base 10 de ce nombre. Cette transformation est robuste aux changements d'illumination (et de la correction gamma des caméras standard). À noter que cette transformation est équivalente (à l'ordre de parcours près) aux codes *Local Binary Pattern (LBP)* décrit par Ojala, Pietikäinen et Harwood [OPH96] qui ont montré de bonnes performances en classification de texture.

Finalement, l'image est découpée en $M \times M$ régions et le descripteur est obtenu en calculant l'histogramme dans chacune des sous-régions. Il y a 256 valeurs possibles pour la CT (car on travaille sur 8 bits). Cependant, deux valeurs proches de la CT peuvent signifier des voisinages complètement différents. Il n'y a donc pas de raison de faire un histogramme comportant moins de classes. Chaque région est donc décrite par un vecteur à 256 dimensions. À noter que Wu ne concatène pas les vecteurs obtenus dans chaque région en un seul vecteur.

Une limitation du descripteur CENTRIST est qu'il ne travaille qu'à une seule échelle. De plus, il existe des détecteurs de contours plus efficaces que le filtre de Sobel. Enfin, les valeurs de la CT en différents pixels sont corrélées. Par exemple, le bit 5 du pixel (x, y) (il vaut 1 si $\mathbf{I}(x, y) \geq \mathbf{I}(x, y + 1)$) et le bit 4 du pixel $(x, y + 1)$ (il vaut 1 si $\mathbf{I}(x, y) \leq \mathbf{I}(x, y + 1)$) sont complémentaires (sauf dans le cas où $\mathbf{I}(x, y) = \mathbf{I}(x, y + 1)$). Wu mentionne que ces contraintes (et d'autres) existent mais ne les exploite pas. Dans [WR08], une ACP est utilisée pour réduire la dimensionnalité du vecteur.

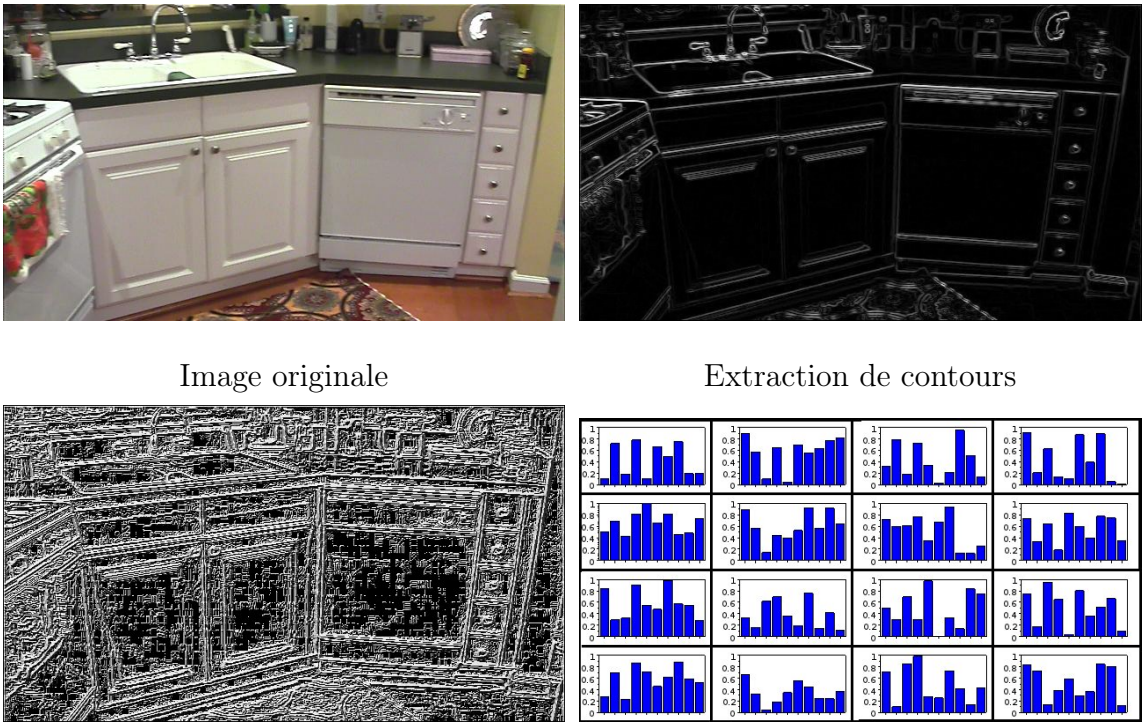
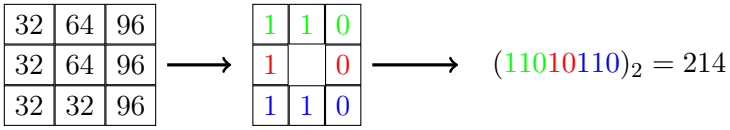


Image originale | Extraction de contours

CT | Histogramme

(a) Étapes du calcul de CENTRIST



(b) Census Transform (CT)

FIGURE I.16.: Descripteur CENTRIST : (a) les étapes du calcul (b) : principe de la CT. Adapté de [Wu09].

2.3. Extraction de zones d'intérêt

Même si nous sommes capables de saisir le sens d'une image globalement, on sait que lorsqu'on présente une image à un sujet (même sans tâche explicite), il explore inconsciemment la scène avec des saccades oculaires. Les points de fixation (appelés points saillants) sont reproductibles entre individus.

Dès lors, on peut postuler que certaines zones sont plus intéressantes que d'autres dans la caractérisation de l'image. La première étape d'une description locale des images est donc de trouver les zones intéressantes. Bien entendu, la notion de zone intéressante dépend de la tâche. Il existe de nombreuses techniques que nous détaillons dans cette section.

2.3.1. Échantillonnage dense

La technique la plus simple pour caractériser localement une image est d'échantillonner les points régulièrement selon une grille, éventuellement à plusieurs échelles. Cette technique est différente de la géométrie grossière évoquée au paragraphe précédent car ici, on n'utilise qu'un voisinage autour des points pour caractériser l'image et non toute une région.

2.3.2. Segmentation

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères pré-définis. Les pixels sont ainsi regroupés en régions qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond. Contrairement au découpage par bloc, la segmentation se fait à partir du contenu visuel.

Il est relativement facile de regrouper des pixels ayant des caractéristiques bas-niveau communes. En revanche, la segmentation de haut-niveau (où chaque région correspond à un objet) est encore difficile. Il existe cependant des techniques de segmentation basées sur l'utilisation de caractéristiques générales des objets (existence de frontières, apparence différente du fond) qui permettent de trouver des objets sans faire appel à des informations sur la classe (voir [ADF10]).

2.3.3. Régions d'intérêt

Une région d'intérêt est une région de l'image dont on considère qu'elle contient beaucoup d'information. Suivant les applications, on s'intéresse :

– Aux régions particulièrement uniformes (souvent appelée *blob* en anglais) selon un certain critère (par exemple, c'est une région qui est soit plus brillante, soit plus sombre que son voisinage). Contrairement au problème de la segmentation, il n'est pas nécessaire que chaque pixel soit attribué à une région : le résultat de l'algorithme est un ensemble de régions non-connexes.

– Aux régions où le signal change beaucoup (bords, coins, contours).

Dans la plupart des cas, l'algorithme détecte des points d'intérêt (caractérisés par une position dans l'image $\boldsymbol{x}_i$) ainsi qu'une petite région autour du point d'intérêt, appelée région locale d'intérêt A_i, son échelle caractéristique σ_i et son orientation caractéristique θ_i. Un détecteur de point d'intérêt est donc une fonction d, qui à une image $\mathbf{I}$ associe une liste de points d'intérêt $\mathrm{d}(\mathbf{I}) = \{(\boldsymbol{x}_i, A_i, \theta_i, \sigma_i), i = 1, \ldots, N_{\mathbf{I}}\}$.

La plupart des détecteurs de points d'intérêt ont été inventés pour le recalage d'images ou pour trouver des correspondances entre deux images d'un même objet prises sous des angles différents (dans le but de construire un modèle 3D de l'objet à partir de plusieurs vues). Partant de deux images $\mathbf{I}$ et $\mathbf{I}'$ issues de la même pré-image, le but est donc de retrouver les mêmes zones de la pré-image ([Mik+05]). Dans la pratique, on ne connaît pas la pré-image mais il existe une certaine transformation f entre les deux images : $\mathbf{I}' = \mathrm{f}(\mathbf{I})$.

On souhaite donc que les régions détectées par le détecteur d dans l'image transformée $\mathrm{f}(\mathbf{I})$ soient les transformées des régions détectées dans $\mathbf{I}$, *i.e.* :

$$\mathrm{d}(\mathrm{f}(\mathbf{I})) = \mathrm{f}(\mathrm{d}(\mathbf{I})) \tag{I.4}$$

Les régions détectées sont donc covariantes puisqu'elles sont transformées en même temps que l'image. Par abus de langage, on les appelle invariantes.

D'un point de vue géométrique, les transformations qu'une image est susceptible de subir et auxquelles les détecteurs doivent être invariants sont les affinités, transformations géométriques qui décrivent le changement de point de vue. Ces transformations regroupent (voir Figure I.17) :

– Les translations : en général, la translation d'une image ne transforme pas son contenu ; la plupart des détecteurs sont basés sur les extrema locaux d'une certaine quantité définie localement donc invariants en translation (notons quand même que, l'image étant finie, une translation fait disparaître du champ visuel une partie de la scène, dont les points d'intérêt sont perdus ; de même les nouvelles parties introduisent des nouveaux points d'intérêt)
– Les changements d'échelle (ou similarités) : l'invariance en échelle signifie que l'on retrouve la même zone dans une image zoomée. La technique de base pour obtenir un détecteur de points d'intérêt invariant en échelle est l'utilisation de la représentation espace-échelle dans lequel une image est convoluée avec des gaussiennes isotropiques d'écart-type croissant. Si l'on calcule la réponse du détecteur à toutes les échelles, pour une position donnée, l'échelle à laquelle la réponse est maximale est caractéristique du point d'intérêt.
– Les changements d'angle de vue : une approche similaire à l'invariance en échelle utilisant des gaussiennes anisotropes est possible mais difficile en pratique ; [MS02] utilise une technique itérative basée sur la matrice des moments d'ordre deux pour calculer la bonne affinité.

– Les rotations : comme pour les translations, la plupart des méthodes sont naturellement insensibles aux rotations car elles sont basées sur des extrema locaux ; afin d'obtenir des régions covariantes avec la rotation, on affecte à chaque région une orientation préférentielle. L'approche la plus stable (proposée par [Low99] et améliorée par [Low04]) se base sur les histogrammes de l'orientation du gradient dans une zone proche du point d'intérêt.

On souhaite aussi que les détecteurs soient robustes aux transformations photométriques globales. La plupart des méthodes utilisent les gradients et sont donc naturellement insensibles aux transformations photométriques ; de plus la position des maxima locaux est, elle aussi, indépendante des transformations photométriques.

Comme nous l'avons dit, la plupart des détecteurs sont basés sur les extrema locaux d'une quantité calculée localement à l'aide d'un filtre. Dans cette optique, un point saillant (extremum local) est un point d'intérêt. Les contours portant l'essentiel de l'information (voir page 29), les points saillants sont en général les extrema locaux d'un détecteur de contours. Les détecteurs basés sur les contours les plus utilisés sont Harris-Affine, Hessian-Affine, Difference of Gaussians (DoG).

Le détecteur Maximally Stable Extremal Region (MSER) permet de détecter des régions d'intérêt covariantes. Les régions détectées sont celles dont la valeur est inférieure à un seuil et qui restent stables quand on augmente ce seuil.

Kadir et Brady [KB01] ; Kadir, Zisserman et Brady [KZB04] ont développé un système de détection de régions d'intérêt basé sur l'entropie. Les régions détectées sont celles qui sont hautement imprédictibles.

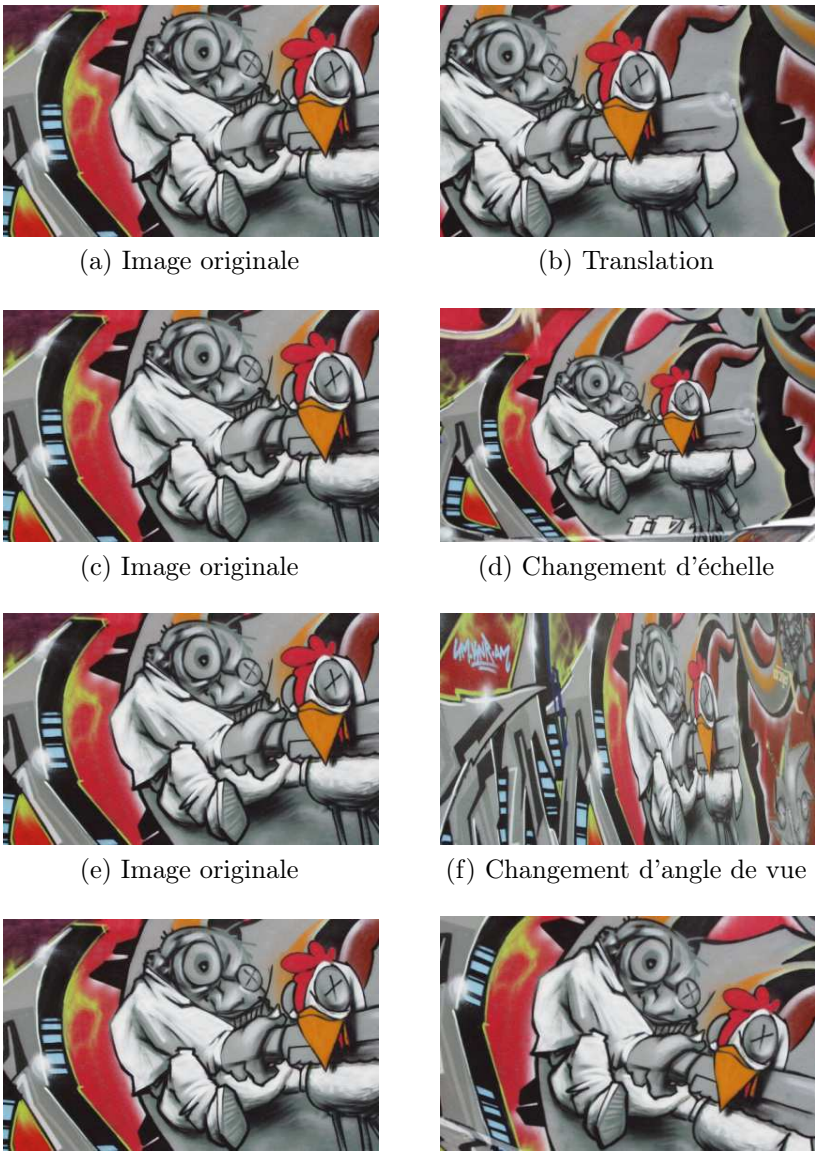
2.3.4. Focalisation attentionnelle

Comme nous l'avons dit, l'idée de la caractérisation locale d'une image est très liée à l'attention visuelle. La référence implicite de la plupart des systèmes de vision étant le système visuel humain, on est enclin à penser que les points de fixation attentionnels sont de bons candidats pour la caractérisation des images. On peut donc chercher à reproduire certains principes de l'attention visuelle.

La notion d'attention recouvre deux aspects complémentaires :

1. L'attention ascendante (encore appelée pré-attention) qui se manifeste par l'attention portée à un changement brutal dans l'environnement. Ces processus sont complètement dirigés par les donnés et, en général, inconscients.
2. L'attention descendante, guidée par la tâche. Par exemple, la recherche active d'un objet rouge dans une scène visuelle peut « biaiser » les mécanismes de focalisation attentionnelle pour porter plus d'attention aux points rouges.

Les systèmes attentionnels bio-inspirés présentent plusieurs avantages :



(a) Image originale (b) Translation

(c) Image originale (d) Changement d'échelle

(e) Image originale (f) Changement d'angle de vue

(g) Image originale (h) Rotation

FIGURE I.17.: Les différentes transformations géométriques d'une image. Adapté de [MS05].

1. Premièrement, et contrairement aux méthodes susmentionnées, notamment aux méthodes basées sur les contours, ces systèmes combinent plusieurs indices pour calculer des points de saillance maximale. En effet, le système visuel humain sépare les entrées en modalités visuelles simples (couleurs, orientation, formes, *etc.*). Les travaux de [TG80] ont montré que des saillances sont calculées en parallèle sur chaque modalité, avant d'être combinées pour détecter des points d'intérêt. La combinaison de modalités différentes confère aux systèmes bio-inspirés une grande souplesse face à un environnement varié et ouvert et évite de faire des hypothèses sur l'importance à accorder à une caractéristique plutôt qu'une autre.
2. Deuxièmement, les systèmes attentionnels bio-inspirés permettent d'ordonner « naturellement » quelques points d'intérêt dans l'image, sous la forme de focus d'attention, là où les systèmes classiques de traitement de l'image proposent généralement des centaines ou des milliers de points d'importance équivalente.
3. Troisièmement, il est possible de biaiser le système pour incorporer des informations liées à la tâche et ce, de manière dynamique. La modélisation de l'attention descendante est compliquée mais on peut néanmoins introduire des connaissances générales sur les mécanismes attentionnels. Par exemple, dans la thèse de Sylvain Chevallier ([Che09]), les saillances à grande échelle vont permettre de pré-activer certaines zones. De plus, l'architecture choisie permet de maintenir la focalisation sur le point saillant détecté au cours du temps. De même, Hervé Guillaume a développé durant sa thèse ([Gui09, chap. 6]) un système attentionnel inspiré des champs neuronaux qui permet de maintenir le point de fixation lors du déplacement d'un robot mobile.

Les premières étapes du système visuel humain sont massivement parallèles, or nous ne pouvons prêter attention qu'à un seul endroit à la fois. Il a été proposé (voir [Tso92]) que le caractère sériel de l'attention n'est pas une limite du système visuel mais permet au contraire de diminuer la complexité (au sens informatique) de la recherche visuelle.

La modélisation la plus employée est sans doute celle issue des travaux de Itti, Koch et Niebur [IKN98]. La Figure I.18 montre l'architecture d'ensemble du système. Les modalités de l'image d'entrée (intensité, couleurs, orientations) sont séparées, à différentes échelles (cette opération est obtenue par un simple filtrage). Sur chaque modalité, des opérateurs de contraste locaux, inspirés des cellules on-off et off-on de la rétine, permettent de calculer des cartes de traits (*feature maps*). Les contrastes locaux sont donc considérés comme des indicateurs pertinents des régions supportant un maximum d'information. Ces cartes sont combinées en une carte globale dite « carte de perceptibilité » (*conspicuity maps*) pour chaque modalité. Les cartes de perceptibilité sont combinées en une carte de saillance.

Un réseau Winner-take-all (WTA) permet de déterminer la région la plus saillante dans la carte, c'est-à-dire le focus d'attention. Le réseau WTA et le détail de son implémentation représentent une importante contribution de ce travail. Il permet notamment d'expliquer

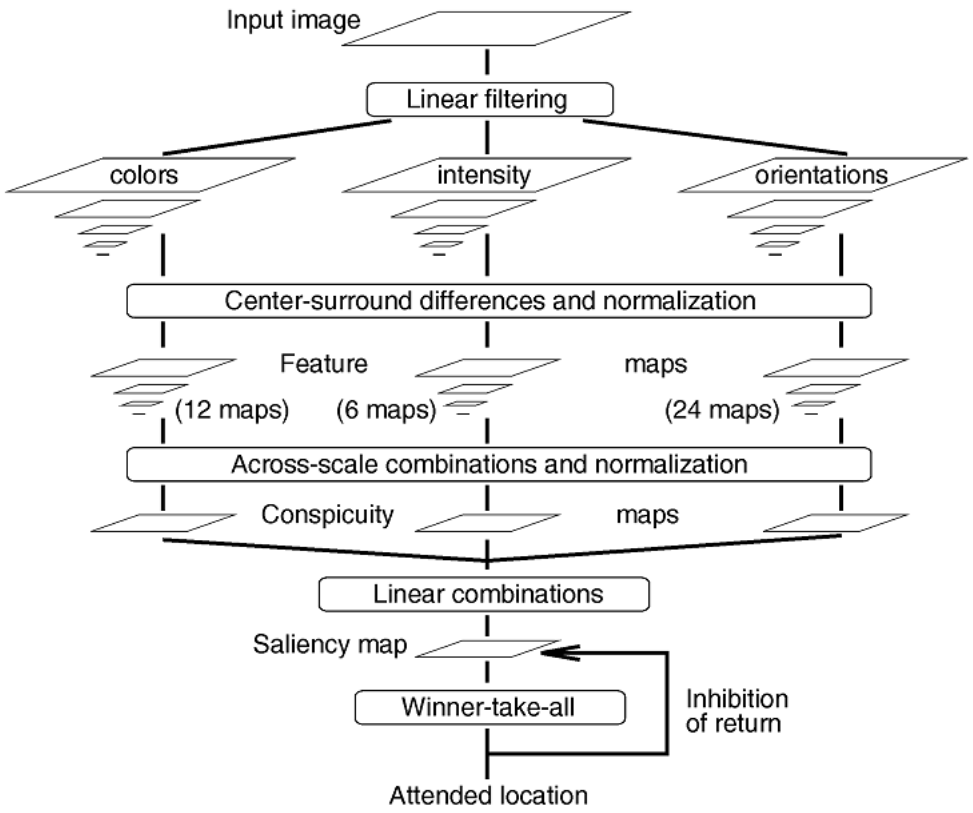


FIGURE I.18.: Le modèle de ITTI, KOCH et NIEBUR [IKN98].

comment la sélection d'un maximum global peut être réalisée par des unités qui sont seulement connectées localement. Après sélection de la région la plus saillante, celle-ci donne lieu à une représentation centrale qui contient les propriétés d'une région unique dans la scène visuelle. Le déplacement du focus d'attention au sein de la carte de saillance est assurée par un mécanisme d'inhibition de retour (Inhibition Of Return : IOR) qui désactive la région sélectionnée et cause un déplacement du focus vers la prochaine zone la plus saillante. Dans son ensemble, l'architecture proposée est simplement ascendante. Pour plus de détails sur les modélisations de l'attention, le lecteur est invité à se tourner vers Chevallier [Che09, chap. 1] ou Guillaume [Gui09, chap. 2].

Les liens entre attention et surprise ont été formalisés grâce aux principes bayésiens. Le lecteur intéressé pourra se rapporter à Itti et Baldi [IB06].

2.4. Caractérisation locale des images

Une fois les zones intéressantes trouvées, on peut les caractériser en utilisant des descripteurs locaux. Ici aussi, il nous est impossible de décrire toutes les alternatives. Le lecteur est invité à consulter [Tir10, chap. 1] pour plus de détails.

Le descripteur local le plus employé est sans doute Scale-Invariant Feature Transform (SIFT) décrit par Lowe [Low99 ; Low04] [18].

2.4.1. SIFT

Le descripteur SIFT est essentiellement un histogramme d'orientation du gradient. Rappelons que le détecteur de points d'intérêt calcule l'échelle et l'orientation dominante de la région : le descripteur est donc indépendant des zooms et des rotations. On extrait une région rectangulaire de taille $M \times M$ autour du point d'intérêt (dans la majorité des cas, $M = 16$; à des fins d'illustration dans la Figure I.19, $M = 8$). On calcule le gradient et on normalise son orientation par rapport à l'orientation dominante de la région. L'amplitude du gradient est pondérée par une fonction gaussienne centrée sur le point d'intérêt.

On réalise ensuite des histogrammes locaux d'orientation dans des régions de taille $N \times N$ (dans la majorité des cas, $N = {}^{M}/_{4}$; à des fins d'illustration dans la Figure I.19, $N = 4$). Les histogrammes comportent 8 classes et les votes sont pondérés par l'amplitude (qui a été elle-même pondérée).

Avec les paramètres standard, la dimension du descripteur est donc $4 \times 4 \times 8 = 128$. Des variantes de SIFT ont été proposées, utilisant la couleur(voir [AF06]) ou réduisant la taille

18. La méthode décrite dans ces articles regroupe à la fois la détection de points d'intérêt invariants par similarité et leur description (ainsi que d'autres techniques). Beaucoup d'auteurs utilisent le descripteur SIFT en conjonction avec un détecteur invariant par affinité. Lorsque nous parlons de SIFT, nous parlons de la description.

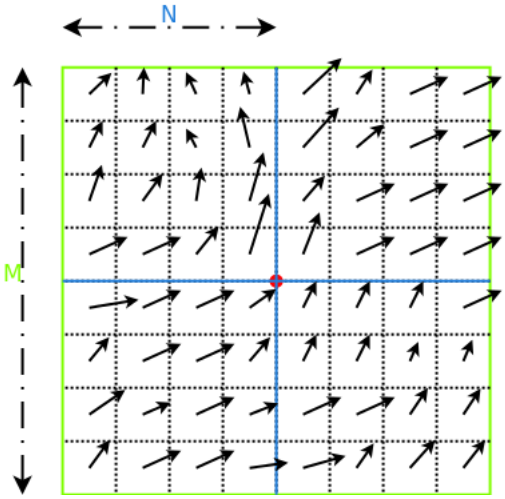


FIGURE I.19.: Descripteur SIFT. Ici $M = 8$, $N = 4$. Le point d'intérêt est au centre. Les pointillés marquent les pixels. Le gradient est calculé sur le fenêtre verte. Les histogrammes sont calculés sur les régions bleues. Ici, la dimension du descripteur serait ${}^{M}/_{N} \times {}^{M}/_{N} \times 8 = 2 \times 2 \times 8 = 32$. Tiré de [Low04].

du descripteur à l'aide de l'ACP (voir [Ke04]).

Parmi les autres descripteurs proches de SIFT, citons Histogram of Oriented Gradients (HOG) (voir [DT05]), lui aussi basé sur l'histogramme des gradients. Récemment, le descripteur Speeded-Up Robust Features (SURF) a été décrit (voir [Bay+08]). Les contours sont décrit à l'aide d'ondelettes de Haar, ce qui permet une implémentation très rapide à l'aide des images intégrales.

2.5. Conclusion

Cette partie a présenté quelques grandes techniques en vision par ordinateur. La question de savoir comment représenter une image en évitant le double écueil de la variabilité et de la confusion perceptuelle conduit au développement de méthodes efficaces de représentation de l'image.

Dans la plupart des tâches classiques en vision par ordinateur, on dispose d'images de référence. Par exemple, dans une tâche de reconnaissance d'objets, on dispose de photos des objets sous différentes conditions (par exemple plusieurs points de vue).

Pour les modèles de représentation simples, les règles de classification des objets peuvent être directement intégrées au système de reconnaissance par un expert. Pour les modèles plus complexes, notamment lorsque les données sont représentées dans un espace de grande dimension et que le nombre de classes est important, il est impossible de fournir manuellement des critères pertinents de classification. Ces critères doivent être appris à partir de données

supervisées, notamment grâce aux méthodes issues de l'apprentissage machine.

3. Apprentissage artificiel

3.1. Introduction

L'apprentissage artificiel (voir Cornuéjols et Miclet [CM10]) est un sous-domaine de l'intelligence artificielle qui vise à l'écriture de programmes capables de s'améliorer automatiquement à partir de données empiriques. La plupart du temps, les données sont disponibles sous forme d'une base de données et le programme construit un modèle en vue de résoudre une certaine tâche.

Quelques tâches classiques de l'apprentissage artificiel sont :
- la classification de données (supervisée ou non) ;
- la régression ;
- l'apprentissage de densité de probabilité ;
- plus récemment, l'apprentissage de fonctions de tri.

Ces techniques ont été appliquées au problème de la reconnaissance de la parole, au contrôle de robot, en vision par ordinateur, *etc.*

Dans le cas où les entrées sont présentées comme un vecteur d'attributs (numériques ou catégoriques), on parle d'apprentissage propositionnel. Dans le cas où les entrées sont structurées (chaîne, graphe, *etc.*), on parle d'apprentissage relationnel. L'apprentissage de séquences en bio-informatique ou en fouille de texte en constituent de bons exemples.

Nous aborderons le problème de la reconnaissance et la catégorisation de lieux sémantiques sous l'angle de la classification supervisée. De plus, les techniques que nous utilisons sont basées sur une étape de classification non-supervisée. Nous présentons successivement ces deux domaines. Nous nous restreignons au cas propositionnel et aux espaces de représentation numériques : les entrées sont des vecteurs issus d'un espace $\mathcal{X}$ de dimension N (en général, $\mathcal{X} \subset \mathbb{R}^N$). Dans le cas des images, $\mathcal{X}$ pourrait être l'espace d'état des images. Cependant, les techniques que nous avons vues dans la section précédente permettent une description plus efficace des images, ce qui facilite le processus d'apprentissage [19].

3.1.1. Classification supervisée

Dans les problèmes supervisés, chaque donnée est associée à une classe connue. On a donc un ensemble fini $\mathcal{C} = \{c^1, \ldots, c^p\}$ de p classes. Chaque exemple est un couple $(\boldsymbol{x}, y)$ où $\boldsymbol{x} \in \mathcal{X}$

19. La sélection d'attributs (*feature selection*) est la branche de l'apprentissage artificiel qui se charge de trouver automatiquement la description la plus pertinente des données d'entrées. Certaines études utilisent des techniques d'apprentissage évoluées pour la sélection d'attributs. Cependant, même dans ce cas, les données sont en général pré-traitées.

est la représentation de l'objet et $y \in \mathcal{C}$ est la classe de $\boldsymbol{x}$ [20]. L'ensemble d'apprentissage $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$ est constitué de m exemples.

L'apprentissage consiste à inférer, à partir de $\mathcal{S}$, une règle permettant d'attribuer une classe à chacun des points de l'espace $\mathcal{X}$. Dans un cadre statistique, établir une règle de classification est équivalent à modéliser la probabilité $\mathrm{P}(y|\boldsymbol{x})$.

On peut séparer le problème en deux tâches distinctes :

- l'étape d'inférence, c'est-à-dire l'estimation de la distribution de probabilité $\mathrm{P}(y|\boldsymbol{x})$
- L'étape de décision, c'est-à-dire le choix de la classe optimale $\overset{\star}{y}$ en fonction de la probabilité *a posteriori* $\mathrm{P}(y|\boldsymbol{x})$

En ce qui concerne l'inférence, on distingue deux grandes familles de méthodes d'apprentissage selon qu'elles apprennent directement ou indirectement $\mathrm{P}(y|\boldsymbol{x})$: les méthodes discriminatives et les méthodes génératives. Nous détaillons plus loin ces méthodes (voir Section I.3.2).

3.1.2. Classification non-supervisée

En apprentissage non-supervisé, les exemples d'apprentissage ne sont pas étiquetés. Le but est donc de trouver les classes naturelles (ou *clusters*) de l'ensemble $\mathcal{X}$ et de grouper les données dans ces classes. Il n'y a pas de définition précise d'une classe naturelle mais habituellement, on considère que deux propriétés doivent être réunies :

1. Les membres d'une même classe doivent être proches, ce qui revient à minimiser la variance intra-classe.
2. Les membres de deux classes doivent être différents, ce qui revient à maximiser la variance inter-classes.

La mesure de la similarité entre les objets, d'une part, et entre les groupes d'objets, d'autre part, apparaissent donc comme des concepts centraux de la classification non-supervisée. Le compromis entre la variance intra-classe et la variance inter-classes peut être vu comme une expression du compromis entre variabilité perceptuelle et confusion perceptuelle : si la variance inter-classes est trop faible (par rapport à la variance intra-classes), on est en situation de confusion perceptuelle ; à l'opposé, si la variance intra-classe est trop petite deux entrées légèrement différentes seront classées différemment (variabilité perceptuelle).

Il existe plusieurs types d'algorithmes :

- Les algorithmes basés sur les densités comme OPTICS (voir [Ank+99]), modélisent directement la densité $\mathrm{p}(\boldsymbol{x})$; une classe est définie comme une région de l'espace dont la densité dépasse un certain seuil. On peut aussi utiliser une modélisation paramétrique de la densité basée sur le mélange de distribution. Le choix le plus courant est d'utiliser un mélange de K gaussiennes (*Gaussian Mixture Model (GMM)*). Chaque exemple est

20. La classe joue le même rôle que l'état caché en dans la Section I.1.4. Nous la notons y car c'est en général une variable scalaire discrète.

affecté à une gaussienne, ce qui permet de définir des groupes. L'algorithme Expectation-Maximization (EM) (voir [DLR77]) permet de déterminer les paramètres des gaussiennes.
- Les algorithmes hiérarchiques construisent une partition de l'espace soit en agglomérant des clusters de plus bas niveau, soit en divisant des groupes en sous-clusters.
- Les algorithmes partitifs séparent les données en plusieurs classes en essayant de minimiser une fonction de coût. L'exemple le plus connu est l'algorithme des K-moyennes (voir [Ste56]). Le nombre de classes est en général fixé mais peut faire partie de la fonction d'erreur.
- Les algorithmes spectraux utilisent le spectre de la matrice de similarité (*i.e.* la matrice des distances entre chaque échantillon)

La plupart des algorithmes ne déterminent pas les clusters directement mais des points de l'espace $\mathcal{X}$ appelés centres des clusters. C'est par exemple le cas de l'algorithme K-moyennes dans lequel chaque « moyenne » correspond à un point de l'espace $\mathcal{X}$. La classe (*i.e.* le cluster) associé au centre $1 \leq k \leq K$ est définie par l'ensemble des points de $\mathcal{X}$ qui sont plus proches du centre k que de n'importe quel autre. Autrement dit, les centres déterminent la partition de Voronoï de $\mathcal{X}$ et un cluster est une cellule de cette partition. Les centres sont appelés prototypes (car ils représentent un exemplaire typique de la classe) ou, par abus de langage, « clusters ».

Une propriété importante d'un algorithme de classification non-supervisée est le degré d'appartenance d'un exemple à une classe :

- dans les algorithmes de classification « nette » (*crisp classification*), chaque point appartient à une classe unique ; c'est le cas dans l'algorithme des K-moyennes.
- dans les algorithmes de classification floue (*fuzzy classification*), un point est associé à une classe via un certain poids (qui peut représenter une probabilité ou un degré d'appartenance en logique floue) ; par exemple, lorsque l'on utilise un mélange de gaussiennes, chaque exemple est assigné à une gaussienne de manière probabiliste.

La plupart des algorithmes nécessitent de préciser le nombre de clusters (le K dans K-moyennes ou le nombre de gaussiennes dans GMM). La plupart des algorithmes qui travaillent sur la densité n'ont pas ce besoin. Parmi les algorithmes partitifs qui déterminent le nombre de clusters, citons *Affinity Propagation* (voir [FD07]).

Nos travaux se basent sur l'algorithme des cartes auto-organisatrices (voir [Koh90]) que nous présentons dans l'Appendice B. Cet algorithme fournit un ensemble de prototypes (appelés « nœuds » dans ce contexte) disposés géométriquement sur un réseau bidimensionnel. Le nombre de clusters est déterminé par le nombre de nœuds de la carte qui est en général fixé (il existe cependant des extensions dans lesquelles on peut ajouter des neurones en cours d'apprentissage).

3.2. Apprentissage génératif et discriminatif

Comme nous l'avons dit, deux approches sont possibles pour la modélisation de la probabilité conditionnelle *a posteriori* $\mathrm{P}(y|\boldsymbol{x})$.

3.2.1. Apprentissage discriminatif et fonction de décision

L'approche discriminative cherche à estimer $\mathrm{P}(y|\boldsymbol{x})$ directement. Dans l'approche de Vapnik (voir [Vap98 ; Vap99]), on cherche en fait une fonction $\mathrm{h} : \mathcal{X} \mapsto \mathcal{C}$ qui est capable de prédire l'étiquette associée à un exemple $\boldsymbol{x}$ (ces fonctions sont appelées « hypothèses »). Dans ce cadre, l'estimation des distributions de probabilité n'est plus importante. L'apprentissage est ramené à un problème d'estimation de fonction : trouver la fonction qui minimise le risque de mauvaise classification. Cela revient à déterminer les frontières de décision (ou fonctions de décision) qui permettront d'attribuer une classe à chacun des points de l'espace des données.

Plus formellement, le scénario est le suivant :

- on tire des vecteurs selon une densité $\mathrm{p}(\boldsymbol{x})$ inconnue mais fixe
- un superviseur assigne une classe $y \in \mathcal{C}$ avec une probabilité conditionnelle $\mathrm{P}(y|\boldsymbol{x})$ elle aussi inconnue et fixe
- un algorithme d'apprentissage permet de parcourir un ensemble d'hypothèses $\mathcal{H}$.

On a une fonction de coût attachée à une hypothèse $\mathrm{L_h} : \mathcal{X} \times \mathcal{C} \mapsto \mathbb{R}$ qui donne le coût de la classification d'un vecteur $\boldsymbol{x}$. En classification, la fonction de coût la plus utilisée est la fonction :

$$\mathrm{L_h}(\boldsymbol{x}, y) = \begin{cases} 0 & \text{si} \quad \mathrm{h}(\boldsymbol{x}) = y \\ 1 & \text{si} \quad \mathrm{h}(\boldsymbol{x}) \neq y \end{cases} \tag{I.5}$$

Le risque réel associé à l'hypothèse h est l'espérance de coût :

$$\mathrm{R_{réel}}(\mathrm{h}) = \int \mathrm{L_h}(\boldsymbol{x}, y)\, \mathrm{dp}(\boldsymbol{x}, y) \tag{I.6}$$

Avec la fonction de coût définie à l' Équation I.5, le risque réel équivaut à la probabilité de mauvaise classification.

Le but de l'apprentissage est de trouver la fonction $\overset{\star}{\mathrm{h}}$ qui minimise le risque :

$$\overset{\star}{\mathrm{h}} = \arg\min_{\mathrm{h} \in \mathcal{H}} \mathrm{R_{réel}}(\mathrm{h}) \tag{I.7}$$

Comme l'estimation du risque réel est impossible (car on ne connaît pas $\mathrm{p}(\boldsymbol{x}, y)$), on utilise en général le risque empirique qui se base sur l'ensemble d'apprentissage $\mathcal{S}$:

$$\mathrm{R_{emp}}(\mathrm{h}) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{L_h}(\boldsymbol{x}_i, y_i) \tag{I.8}$$

Le principe de minimisation du risque empirique (*Empirical Risk Minimization (ERM)*) consiste à remplacer $\overset{\star}{\mathrm{h}}$ par l'hypothèse $\widehat{\mathrm{h}}$ qui minimise le risque empirique :

$$\widehat{\mathrm{h}} = \underset{\mathrm{h}\in\mathcal{H}}{\arg\min}\, \mathrm{R}_{\mathrm{emp}}\left(\mathrm{h}\right) \tag{I.9}$$

Une frontière de décision permet de séparer l'espace $\mathcal{X}$ en deux classes. Par exemple, une frontière de décision est décrite par un vecteur directeur $\boldsymbol{\omega}$ et un biais b. La fonction de décision associée est donnée par le signe de la distance algébrique à l'hyperplan (l'orientation est arbitraire) :

$$\widehat{\mathrm{h}}\left(\boldsymbol{x}\right) = \mathrm{sgn}\left(\langle\boldsymbol{\omega}, \boldsymbol{x}\rangle + b\right)$$

Il existe plusieurs manières d'étendre les frontières de décisions pour les problèmes multi-classes, dont les plus courantes sont :

- on peut construire construire une frontière permettant de séparer la classe c^1 de c^2, puis une frontière permettant de séparer c^1 de c^3, *etc.* Au total, on doit donc construire $\frac{p\times(p-1)}{2}$ frontières. Cette méthode est appelée *one-against-one*. Pour classer un exemple, on utilise le plus souvent un système de vote : chaque frontière vote pour une classe ; après avoir recueilli les $\frac{p\times(p-1)}{2}$ votes, la réponse finale est la classe désignée le plus souvent.
- on peut aussi construire une frontière permettant de séparer c^1 de toutes les autres classes, puis une frontières permettant de séparer c^2 de toutes les autres classes, *etc.* Au total, on doit donc construire p frontières. Cette méthode est appelée *one-against-all*. Pour classer un exemple, on calcule la distance de $\boldsymbol{x}$ à chaque hyperplan : la classe choisie est celle dont la distance à l'hyperplan séparateur est la plus grande.

Les approches discriminatives sont à la base des méthodes de classification reposant sur l'apprentissage de surfaces séparatrices, tels les Support Vector Machines (SVM) (voir [CV95 ; BGV92]) et une partie des modèles connexionnistes. Ces méthodes se concentrent sur les règles de décision en s'appuyant sur les traits discriminants entre classes.

3.2.2. Apprentissage génératif

La deuxième approche consiste à utiliser la règle de Bayes pour exprimer la probabilité inconnue en fonction de ce qui est connu :

$$\mathrm{P}\left(y|\boldsymbol{x}\right) = \frac{\mathrm{p}\left(\boldsymbol{x}|y\right)\times\mathrm{P}\left(y\right)}{\mathrm{p}\left(\boldsymbol{x}\right)} \tag{I.10}$$

Cette équation fait intervenir trois termes :

- La vraisemblance $\mathrm{p}\left(\boldsymbol{x}|y\right)$ est une densité de probabilité maintenue sur l'espace d'entrée $\mathcal{X}$. Intuitivement, elle représente la probabilité d'observer les différentes données dans $\mathcal{X}$ pour chacune des classes appartenant à $\mathcal{C}$. C'est un modèle de la classe y.

– La probabilité *a priori* $\mathrm{P}(y)$ représente la probabilité qu'un exemple appartienne à une classe, indépendamment de l'observation.
– La densité marginale $\mathrm{p}(\boldsymbol{x})$ est la densité de probabilité des données sur $\mathcal{X}$. Elle nous renseigne sur la probabilité d'observer les différents exemples dans l'absolu.

Les méthodes génératives nécessitent donc d'estimer la vraisemblance des données $\mathrm{p}(\boldsymbol{x}|y)$ et les probabilités *a priori* $\mathrm{P}(y)$. En effet, la densité marginale $\mathrm{p}(\boldsymbol{x})$ est indépendante des classes et n'a donc pas d'influence dans le processus de classification. Elle assure uniquement un rôle de normalisation. Ce terme est en général obtenu grâce à la formule des probabilités totales :

$$\mathrm{p}(\boldsymbol{x}) = \sum_{y \in \mathcal{C}} \mathrm{p}(\boldsymbol{x}|y) \times \mathrm{P}(y)$$

Inférence La partie la plus difficile est d'estimer la vraisemblance des données. Cela revient à bâtir, pour chaque classe, une distribution de probabilité dans l'espace de représentation $\mathcal{X}$. On distingue deux approches pour l'apprentissage des densités de probabilités [21] :

1. Les approches paramétriques : on suppose que les densités de probabilité suivent une certaine loi de distribution (par exemple, une loi de distribution gaussienne). Ces lois sont représentées par des fonctions, paramétrées par un vecteur $\boldsymbol{\theta}$ (dans le cas d'une gaussienne, les paramètres sont le vecteur moyenne $\boldsymbol{\mu}$ et la matrice de covariance $\boldsymbol{\Sigma}$). L'apprentissage consiste à estimer les paramètres $\boldsymbol{\theta}$ à partir de l'ensemble des données. Un autre exemple est l'utilisation d'une combinaison linéaire de distributions paramétriques (*e.g.* une combinaison de distributions gaussiennes). L'algorithme EM (voir [DLR77]) est souvent utilisé dans ce cas.

2. Les approches non-paramétriques : on ne suppose aucune contrainte sur la forme des densités de probabilité. Les densités sont estimées localement en observant les exemples d'apprentissage. On suppose que les distributions sont localement régulières. Les fenêtres de Parzen et l'algorithme des K-plus-proches-voisins sont utilisés pour régulariser les probabilités et estimer les densités. Dans le cas où l'espace de représentation des données est discret, il est possible d'estimer directement les distributions avec un histogramme.

Les probabilités *a priori* peuvent être estimées de différentes manières : nous pouvons les considérer équiprobables et appliquer la règle du maximum de vraisemblance ou les apprendre directement à partir de l'ensemble d'apprentissage.

21. Il existe des méthodes semi-paramétriques dans lesquelles une estimation paramétrique et une estimation non-paramétrique sont combinées.

Règle de décision La théorie bayésienne de la décision prescrit de minimiser la perte moyenne :

$$\overset{\star}{y} = \arg\min_{y \in \mathcal{C}} \sum_{y' \in \mathcal{C}} \mathrm{L}\left(y, y'\right) \mathrm{P}\left(y' | \boldsymbol{x}\right) \tag{I.11}$$

Notons qu'ici, contrairement à l'approche discriminative, on ne cherche pas une fonction de décision mais on effectue une décision ponctuelle (c'est-à-dire pour chaque vecteur $\boldsymbol{x}$).

Si on considère que les coûts d'erreur en classification sont égaux, la théorie bayésienne de la décision prescrit de minimiser la probabilité d'erreur en classification, c'est-à-dire de chercher la classe $\overset{\star}{y}$ qui maximise la probabilité *a posteriori* (règle du maximum *a posteriori*) :

$$\overset{\star}{y} = \arg\max_{y \in \mathcal{C}} \left\{\mathrm{p}\left(\boldsymbol{x} | y\right) \times \mathrm{P}\left(y\right)\right\} \tag{I.12}$$

Si l'on suppose que toutes les classes sont équiprobables on obtient la règle de décision suivante (parfois appelée règle du maximum de vraisemblance) :

$$\overset{\star}{y} = \arg\max_{y \in \mathcal{C}} \left\{\mathrm{p}\left(\boldsymbol{x} | y\right)\right\} \tag{I.13}$$

Si on suppose que les attributs des données sont indépendants, on peut décomposer la vraisemblance des données :

$$\mathrm{p}\left(\boldsymbol{x} | y\right) = \prod_{j=1}^{N} \mathrm{p}\left(x_j | y\right)$$

On obtient alors la règle du Classificateur Bayésien Naïf (CBN) :

$$\overset{\star}{y} = \arg\max_{y \in \mathcal{C}} \left\{\prod_{j=1}^{N} \mathrm{p}\left(x_j | y\right) \times \mathrm{P}\left(y\right)\right\} \tag{I.14}$$

Dans le cas où les classes sont équiprobables :

$$\overset{\star}{y} = \arg\max_{y \in \mathcal{C}} \left\{\prod_{j=1}^{N} \mathrm{p}\left(x_j | y\right)\right\} \tag{I.15}$$

La théorie de la décision bayésienne fournit donc des critères qui permettent de classer les données, à condition de connaître leur densité de probabilité conditionnellement aux classes et éventuellement de connaître les probabilités *a priori* de celles-ci.

Ces approches sont appelées génératives car elles permettent de modéliser la distribution jointe $\mathrm{p}(\boldsymbol{x}, y)$. En effet, par définition :

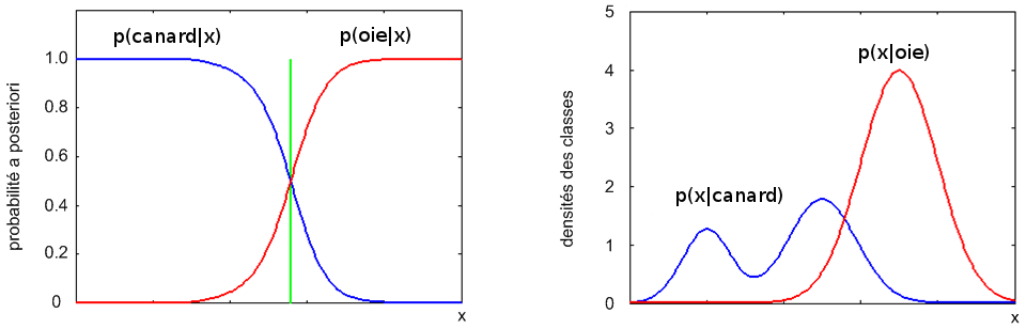
$$\mathrm{p}(\boldsymbol{x}, y) = \mathrm{P}(y|\boldsymbol{x}) \times \mathrm{p}(\boldsymbol{x}) \tag{I.16}$$

On peut dès lors tirer (générer) de nouveaux exemples selon cette loi.

De nombreux classificateurs reposent sur cette approche. Les plus connus et les plus simples sont l'Analyse Discriminante Linéaire (ADL) (ou *Fisher Discriminant Analysis (FDA)*) et le Classificateur Bayésien Naïf. L'approche générative est largement utilisée en reconnaissance visuelle.

3.2.3. Comparaison des approches discriminatives et génératives

La différence entre les méthodes discriminatives et générative peut être illustrée à partir d'un exemple dans le cas simple d'un espace $\mathcal{X}$ à une dimension. Supposons que l'on cherche à différencier les oies et les canards en fonction du poids noté x. On dispose donc d'un ensemble d'observations L'approche discriminative va chercher une fonction (par exemple un hyperplan de $\mathbb{R}^N$) qui sépare au mieux les données. Dans notre exemple, une telle fonction est simplement une valeur de seuil $\widehat{x}$. Ceci est illustré par la ligne verte de la figure I.20a. Quand un exemple se présente, on calcule de quel côté de cette surface il se trouve, ce qui donne directement sa classe. Dans notre exemple, on compare le poids du volatile qu'on cherche à classer à $\widehat{x}$: si celui-ci est inférieur à $\widehat{x}$, c'est un canard, sinon, c'est une oie.



(a) Apprentissage discriminatif (b) Apprentissage génératif

FIGURE I.20.: L'apprentissage discriminatif (a) cherche à apprendre directement une règle de classification. L'apprentissage génératif (b) apprend les densités de probabilité $\mathrm{p}(x|y)$ de chaque classe sur l'espace de représentation. La décision est obtenue dans un deuxième temps grâce la théorie de la décision bayésienne.

L'approche générative va procéder autrement. Dans un premier temps, nous allons utiliser uniquement les données correspondant aux oies pour produire un modèle des oies, c'est-à-dire

estimer $\mathrm{p}(x|c = \text{oie})$. Ceci est illustré par la courbe rouge de la figure I.20b. On fait de même pour les canards (courbe bleue de la figure I.20b). Quand un exemple se présente, on calcule la vraisemblance selon chaque classe *i.e.* les probabilités $\mathrm{p}(x|c = \text{oie})$ et $\mathrm{p}(x|c = \text{canard})$. L'exemple est classé en utilisant la règle MAP (Équation I.12). Les approches génératives apprennent donc de manière indirecte les frontières de décision.

Les méthodes discriminatives, en particulier l'approche par détermination directe des fonctions de décisions, sont en général plus simples et plus flexibles que les méthodes génératives. En effet, on ne cherche pas à modéliser les densités $\mathrm{p}(\boldsymbol{x}, y)$ et $\mathrm{p}(\boldsymbol{x}|y)$.

De plus, les performances en classification obtenues par les approches discriminatives sont généralement supérieures à celles obtenues avec les approches génératives (voir [NJ02]).

Enfin, la règle de décision dans l'approche générative dépend fortement des hypothèses permettant de modéliser $\mathrm{p}(\boldsymbol{x}, y)$. De ce point de vue, les méthodes par détermination directe de la fonction de décision sont plus robustes.

En revanche, les méthodes génératives disposent de deux avantages que nous détaillons dans la suite : la capacité à apprendre à partir de données faiblement supervisées et la vitesse de convergence de l'apprentissage.

Apprentissage faiblement supervisé à partir d'exemples positifs Dans les méthodes basées sur l'apprentissage discriminatif, les frontières de décision sont en quelque sorte établies par répulsion mutuelle. Ainsi, il est nécessaire d'utiliser tous les exemples pour déterminer une frontière de décision. En particulier, dans le cas d'une classification binaire, on doit utiliser les exemples positifs et négatifs pour tracer la frontière. L'introduction d'une nouvelle classe oblige à « redessiner » l'intégralité des frontières de décision dans l'espace de représentation. De plus, on doit disposer d'une étiquette pour chaque exemple.

Par opposition, dans les approches génératives la construction du modèle de classe utilise uniquement les exemples de cette classe (uniquement à partir des exemples positifs dans le cas d'une classification binaire). De plus, les méthodes génératives sont capables d'apprendre même avec des exemples faiblement supervisés (voir [NJ02]).

En robotique, la nature ouverte de l'environnement invite à choisir des méthodes incrémentales et faiblement supervisées. En effet, les exemples négatifs ne sont pas toujours faciles à obtenir. Par exemple, si la tâche est de distinguer les obstacles des endroits où le robot peut naviguer, l'approche discriminative requiert de présenter des exemples de chaque obstacle et de régions sans obstacles. Cela veut dire que le robot ne peut commencer à se mouvoir que lorsque suffisamment d'obstacles lui auront été présentés. Par opposition, avec une approche générative, le robot peut commencer à se mouvoir dans une zone sans obstacles pour apprendre un modèle de ces zones. Si un obstacle se présente, la vraisemblance du modèle appris sera faible et le robot pourra déclencher un apprentissage des zones occupées par un obstacle.

Taille de l'échantillon d'apprentissage Bien que les classificateurs génératifs atteignent de moins bonnes performances asymptotiques (*i.e.* quand la taille de l'échantillon d'apprentissage m tend vers $+\infty$), Ng et Jordan [NJ02] ont montré que les classificateurs génératifs convergent beaucoup plus vite. Ainsi, avec un petit échantillon d'apprentissage, les classificateurs génératifs peuvent se révéler meilleurs.

Nous avons vu que dans le cas du maximum de vraisemblance, les connaissances *a priori* jouent un rôle important dans l'obtention des règles de classification. Leur poids est d'autant plus important que l'échantillon d'apprentissage est réduit. Sous condition que les probabilités *a priori* soient bien informées, une classification raisonnable est possible dans des conditions où les modèles discriminatifs sont intrinsèquement limités. En robotique autonome, où les exemples sont collectés au cours de l'exploration, la taille de l'échantillon d'apprentissage est souvent beaucoup plus réduite que celle des banques d'images préétablies. L'incorporation de connaissances *a priori* peut alors se révéler très utile.

Pour conclure, notons que les approches discriminatives et génératives ne sont pas antinomiques. Par exemple, Ng et Jordan [NJ02] et Bouchard et Triggs [BT04] les présentent de manière unifiée. Si l'on veut modéliser $\mathrm{p}(\boldsymbol{x}, y)$ par une famille de distribution de probabilités paramétrisées par $\boldsymbol{\theta}$, on peut chercher :

- le paramètre qui s'adapte le mieux à la densité jointe $\mathrm{p}(\boldsymbol{x}, y)$
- le paramètre qui s'adapte le mieux à la densité conditionnelle $\mathrm{P}(y|\boldsymbol{x})$
- le paramètre qui minimise l'erreur de classification

La première stratégie revient à faire un modèle génératif et les deux autres, un modèle discriminatif. On appelle « paire discriminative-générative », les modèles qui se basent sur la même famille de distributions. Un exemple de paire discriminative-générative dans le cas des espaces d'entrée $\mathcal{X}$ discrets est le Classificateur Bayésien Naïf et la régression logistique.

Il existe d'ailleurs des approches récentes qui combinent les avantages des deux méthodologies par exemple le noyau de Fischer ([JH99]).

3.3. Conclusion

Les techniques issues de l'apprentissage ont permis de rendre les méthodes de vision plus robustes. En retour, les applications en vision ont stimulé les nouvelles applications de l'apprentissage.

Dans la cadre de la robotique autonome où nous sommes confrontés à un environnement ouvert, plusieurs arguments plaident en faveur des modèles génératifs pour l'apprentissage, notamment la nécessité de disposer de modèles incrémentaux.

4. Conclusion générale

Ce chapitre a présenté les principales disciplines auxquelles nous nous rattachons. La notion de probabilité, et en particulier les approches bayésiennes, a été introduite dans trois cadres :

- en robotique, les approches probabilistes reposent sur l'algorithme de filtrage bayésien (voir page 19) ;
- en vision par ordinateur, la théorie bayésienne permet naturellement de modéliser la perception comme un processus d'inférence et de tirer partie des connaissances apprises dans un domaine pour appuyer la reconnaissance dans un autre ;
- enfin en apprentissage, dans le cadre de la robotique, les méthodes génératives apparaissent les plus adaptées.

Ces trois approches sont intimement reliées. Dans ces conditions, la conception d'un système robotique peut être réalisée à partir de briques individuelles qu'il sera plus facile de combiner en un système de complexité croissante.

Bibliographie

Robotique cognitive

[AMF99] A. Arleo, J. d. R. Millán et D. Floreano. « Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation ». Dans : *IEEE Transactions on Robotics and Automation* 15 (1999), 990–1000 (cf. p. 16).

[Ang+08] A. Angeli et al. « Incremental vision-based topological SLAM ». Dans : *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*. 2008 (cf. p. 22).

[Arl+00] A. Arleo et al. « Place Cells and Spatial Navigation based on Vision, Path Integration, and Reinforcement Learning ». Dans : *Advances in Neural Information Processing Systems 13*. T. 13. Denver, Co : MIT Press, 2000, p. 89–95 (cf. p. 16).

[Bod06] M. A. Boden. *Mind as Machine*. Oxford, England : Oxford University Press, 2006 (cf. p. 13).

[Bra84] V. Braitenberg. *Vehicles : Essays in synthetic psychology*. Cambridge, MA : The MIT Press, 1984 (cf. p. 8, 15).

[Bro86] R. A. Brooks. « A Robust Layered Control System for a Mobile Robot ». Dans : *IEEE Journal of Robotics and Automation* RA-2.1 (1986), p. 14–23 (cf. p. 15, 16).

[Bro90] R. A. Brooks. « Elephants don't play chess ». Dans : *Robotics and Autonomous Systems* 6 (1990), p. 3–15 (cf. p. 15).

[Bro91a] R. A. Brooks. « Intelligence without representation ». Dans : *Artificial Intelligence* 47 (1991), p. 139–159 (cf. p. 15).

[Bro91b] R. A. Brooks. « The Role of Learning in Autonomous Robots ». Dans : *Proceedings of the Fourth Annual Workshop on Computational Learning Theory (COLT 1991)*. Sous la dir. de L. G. Valiant et M. K. Warmuth. Santa Cruz, California, USA : Morgan Kaufmann, 1991, p. 5–10 (cf. p. 15).

[Cas+99] J. A. Castellanos et al. « The SPmap : A probabilistic framework for simultaneous localization and map building ». Dans : *IEEE Transactions on Robotics and Automation* 15.5 (1999), 948–953 (cf. p. 22).

[CL85] R. Chatila et J.-P. Laumond. « Position referencing and consistent world modelling for mobile robots ». Dans : *IEEE International Conference on Robotics and Automation*. 1985, 138–170 (cf. p. 22).

[CQG07] N. Cuperlier, M. Quoy et P. Gaussier. « Neurobiologically inspired mobile robot navigation and planning ». Dans : *Frontiers in NeuroRobotics* 1 (2007) (cf. p. 16, 17).

[DMS99] G. Dedeoglu, M. Mataric et G. S. Sukhatme. « Incremental, online topological map building with a mobile robot ». Dans : *Mobile Robots XIV*. 1999, 129–139 (cf. p. 22).

[Fil09] D. Filliat. *Robotique Mobile*. 2009 (cf. p. 6, 7, 10, 11).

[FM00] M. O. Franz et H. A. Mallot. « Biomimetic robot navigation ». Dans : *Robotics and Autonomous Systems* 30 (2000), p. 133–153 (cf. p. 6, 16).

[Gal+05] C. Galindo et al. « Multi-hierarchical semantic maps for mobile robotics ». Dans : *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE. 2005, p. 2278–2283 (cf. p. 11).

[GC96] C. Gallistel et A. Cramer. « Computations on metric maps in mammals : Getting oriented and choosing a multi-destination route ». Dans : *Journal of Experimental Biology* 199.1 (1996). Journal of Experimental Biology, p. 211–217 (cf. p. 9).

[HBC96] P. Hébert, S. Betgé-Brezetz et R. Chatila. « Decoupling odometry and exteroceptive perception in building a global world map of a mobile robot : The use of local maps ». Dans : *IEEE International Conference on Robotics and Automation*. 1996 (cf. p. 22).

[HP91] R. Hartley et F. Pipitone. « Experiments with the subsumption architecture ». Dans : *Proceedings of the 1991 IEEE International Conference onRobotics and Automation (ICRA 1991)*. IEEE. 1991, p. 1652–1658 (cf. p. 17).

[KW95] D. Kortenkamp et T. Weymouth. « Topological mapping for mobile robots using a combination of sonar and vision sensing ». Dans : *Proceedings of the National Conference on Artificial Intelligence*. JOHN WILEY & SONS LTD. 1995, p. 979–979 (cf. p. 22).

[LDC92] J. J. Leonard, H. F. Durrant-Whyte et I. J. Cox. « Dynamic map building for an autonomous mobile robot ». Dans : *International Journal of Robotics Research* 11.4 (1992), 89–96 (cf. p. 22).

[LL90] T. S. Levitt et D. T. ; Lawton. « Qualitative navigation for mobile robots ». Dans : *Artificial Intelligence* 44.3 (1990), p. 305–360 (cf. p. 6).

[Mat02] M. J. Mataric. « Situated robotics ». Dans : *Encyclopedia of Cognitive Sciences*. Sous la dir. de L. Nadel. Nature Publishers Group, Macmillan Reference Ltd, 2002 (cf. p. 15).

[Mat90] M. J. Mataric. « Navigating with a Rat Brain : A Neurobiologically-Inspired Model for Robot Spatial Representation ». Dans : *First International Conference on Simulation of Adaptive Behavior*. Sous la dir. de J. Meyer et S. Wilson. Cambridge, MA : The MIT Press, 1990, p. 169–175 (cf. p. 16).

[ME85] H. Moravec et A. Elfes. « High resolution maps from wide angular sensors ». Dans : *IEEE International Conference On Robotics and Automation*. 1985 (cf. p. 21).

[Mon03] M. Montemerlo. « FastSLAM : A factored solution to the simultaneous localization and mapping problem with unknown data association ». Thèse de doct. Pisstburgh, PA : Carnegie Mellon university, 2003 (cf. p. 22).

[NH08] A. Nüchter et J. Hertzberg. « Towards semantic maps for mobile robots ». Dans : *Robotics and Autonomous Systems* 56.11 (2008), p. 915–926 (cf. p. 11).

[Nil84] N. Nilsson. *Shakey the robot*. Rap. tech. SRI, 1984 (cf. p. 14).

[OC78] J. O'Keefe et D. Conway. « Hippocampal place units in the freely moving rat : Why they fire where they fire ». Dans : *Experimental Brain Research* 31 (1978), p. 573–590 (cf. p. 9).

[ON78] J. O'Keefe et L. Nadel. *The Hippocampus as a Cognitive Map*. Oxford : Oxford University Press, 1978 (cf. p. 9).

[Pro11] A. Pronobis. « Semantic Mapping with Mobile Robots ». Thèse de doct. Stockholm, Sweden : Royal Institute of Technology (KTH), juin 2011. isbn : ISBN 978-91-7501-039-7 (cf. p. 11, 12).

[PS99] R. Pfeifer et C. Scheier. *Understanding intelligence*. Cambridge, Massachusetts : MIT Press, 1999, p. 697 (cf. p. 15, 17).

[RD09] A. Ranganathan et F. Dellaert. « Bayesian surprise and landmark detection ». Dans : *Proceedings of the IEEE International Conference on Robotics and Automation, 2009 (ICRA 2009)*. IEEE, 2009, p. 2017–2023 (cf. p. 10).

[Ste90] L. Steels. « Towards a Theory of Emergent Functionality ». Dans : *First International Conference on Simulation of Adaptive Behavior*. Sous la dir. de J. Meyer et S. Wilson. Cambridge, MA : The MIT Press, 1990, p. 451–461 (cf. p. 13).

[TBF00] S. Thrun, W. Burgard et D. Fox. « A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping ». Dans : *IEEE International Conference on Robotics and Automation*. 2000 (cf. p. 22).

[TBF05] S. Thrun, W. Burgard et D. Fox. *Probabilistic Robotics*. Intelligent Robotics & Autonomous Agents. MIT Press Ltd, 2005, p. 668 (cf. p. 9, 17, 21).

[Tho+88] C. Thorpe et al. « Vision and navigation for the Carnegie-Mellon Navlab ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.3 (1988), p. 362–373 (cf. p. 16).

[Thr01] S. Thrun. « Is robotics going statistics ? The field of probabilistic robotics ». Dans : *Communications of the ACM* (2001) (cf. p. 17).

[Thr02] S. Thrun. « Particle Filters in Robotics ». Dans : *Proceedings of Uncertainty in AI (UAI)*. 2002 (cf. p. 20).

[Thr98a] S. Thrun. « Bayesian Landmark Learning for Mobile Robot Localization ». Dans : *Machine Learning* 33.1 (1998), p. 41–76 (cf. p. 10).

[Thr98b] S. Thrun. « Learning metric-topological maps for indoor mobile robot navigation ». Dans : *Artificial Intelligence* 99.1 (fév. 1998), 21–71 (cf. p. 22).

[Tol48] E. Tolman. « Cognitive maps in rat and men ». Dans : *The Psychological Review* 55.4 (1948), p. 189–208 (cf. p. 9).

[Tru+97] O. Trullier et al. « Biologically based artificial navigation systems : review and prospects ». Dans : *Progress in Neurobiology* 51 (1997), p. 483–544 (cf. p. 6, 8).

[Tur+88] M. Turk et al. « VITS-A vision system for autonomous land vehicle navigation ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.3 (1988), p. 342–361 (cf. p. 16).

[Win72] T. Winograd. *Understanding natural language*. New York : Academic Press, 1972, p. 191 (cf. p. 13).

Vision par ordinateur

[ADF10] B. Alexe, T. Deselaers et V. Ferrari. « What is an object ? » Dans : *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* (2010), p. 73–80 (cf. p. 40).

[AF06] A. E. Abdel-Hakim et A. A. Farag. « CSIFT : A SIFT Descriptor with Color Invariant Characteristics ». Dans : *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2006), p. 1978–1983 (cf. p. 46).

[Att54] F. Attneave. « Some informational aspects of visual perception ». Dans : *Psychological Review* 61.3 (1954), p. 183–193 (cf. p. 30).

[Bar61] H. Barlow. « Possible principles underlying the transformation of sensory messages ». Dans : *Sensory Communication*. Sous la dir. de W. Rosenblith. Cambridge, MA : The MIT Press, 1961, p. 217–234 (cf. p. 30).

[Bay+08] H. Bay et al. « Speeded-Up Robust Features (SURF) ». Dans : *Computer Vision and Image Understanding* 110.3 (2008), p. 346–359 (cf. p. 47).

[BS95] A. Bell et T. Sejnowski. « An information-maximization approach to blind separation and blind deconvolution ». Dans : *Neural computation* 7.6 (1995), p. 1129–1159 (cf. p. 31).

[BS97] A. Bell et T. Sejnowski. « The "independent components" of natural scenes are edge filters ». Dans : *Vision Research* 37.23 (1997). Vision Research, p. 3327–3338 (cf. p. 31).

[Che09] S. Chevallier. « Implémentation d'un système préattentionnel avec des neurones impulsionnels ». Thèse de doct. Université Paris Sud 11, Orsay : Département d'Informatique, 2009 (cf. p. 44, 46).

[Com94] P. Comon. « Independent Component Analysis : a new concept ». Dans : *Signal Processing* 36.3 (1994), p. 287–314 (cf. p. 31).

[Dau09] J. Daugman. « Computer vision ». 2009 (cf. p. 34, 35).

[Dau80] J. Daugman. « Two-dimensional analysis of cortical receptive field profiles ». Dans : *Vision Research* 20 (1980), p. 447–456 (cf. p. 30).

[Des03] T. Deselaers. « Features for Image Retrieval ». Mém.de maîtr. Aachen, Germany : Computer science department, Aachen, 2003 (cf. p. 29).

[Dou+09] M. Douze et al. « Evaluation of GIST descriptors for web-scale image search ». Dans : *Proceedings of the 8th ACM International Conference on Image and Video Retrieval (CIVR 2009)*. Sous la dir. de S. Marchand-Maillet et Y. Kompatsiaris. Santorini Island, Greece : ACM, juil. 2009. isbn : 978-1-60558-480-5 (cf. p. 36).

[DT02] N. Denquive et P. Tarroux. « Multi-resolution codes for scene categorization ». Dans : *European Symposium on Artificial Neural Networks ESANN 2002*. Sous la dir. de M. Verleysen. Bruges, Be, 2002, p. 281–287 (cf. p. 36).

[DT05] N. Dalal et B. Triggs. « Histograms of Oriented Gradients for Human Detection ». Dans : *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005 (cf. p. 47).

[Fie87] D. Field. « Relations between the statistics of natural images and the response properties of cortical cells ». Dans : *Journal of the Optical Society of America A* 4 (1987), p. 2379–2394 (cf. p. 30).

[Fie94] D. Field. « What is the goal of sensory coding ? » Dans : *Neural Computation* 6 (1994), p. 559–601 (cf. p. 30, 31).

[Fie99] D. Field. « Wavelets, vision and the statistics of natural scenes ». Dans : *Philosophical Transactions of the Royal Society of London. Series A : Mathematical, Physical and Engineering Sciences* 357.1760 (1999), p. 2527–2542 (cf. p. 31).

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 44, 46).

[Hel67] H. v. Helmoltz. *Handbuch der physiologischen Optik*. Voss, Hamburg, 1867 (cf. p. 25).

[Hér10] J. Hérault. « Progress in Neural Processing ». Dans : *Vision : Images, Signals and Neural Networks : Models of Neural Processing in Visual Perception*. T. 19. World Scientific, mar. 2010, p. 308 (cf. p. 30).

[HO00] A. Hyvärinen et E. Oja. « Independent component analysis : algorithms and applications ». Dans : *Neural Networks* 13 (2000), p. 411–430 (cf. p. 31).

[HR04] P. Howarth et S. Rüger. « Evaluation of Texture Features for Content-Based Image Retrieval ». Dans : *Lecture Notes in Computer Science* 3115 (2004), p. 2134–2135 (cf. p. 36).

[HSD73] R. M. Haralick, K. Shanmugam et I. Dinstein. « Textural Features for Image Classification ». Dans : *IEEE Transactions on Systems, Man and Cybernetics* 3.6 (nov. 1973), p. 610–621 (cf. p. 29).

[HW62] D. Hubel et T. Wiesel. « Receptive fields, binocular interactions, and functional architecture in the cat's visual cortex ». Dans : *Journal of Physiology, London* 160 (jan. 1962), p. 106–154 (cf. p. 30).

[Hyv99] A. Hyvärinen. « Fast and robust fixed-point algorithms for independent component analysis ». Dans : *Neural Networks, IEEE Transactions on* 10.3 (1999), p. 626–634 (cf. p. 31).

[IB06] L. Itti et P. Baldi. « Bayesian surprise attracts human attention ». Dans : *Advances in neural information processing systems* 18 (2006), p. 547 (cf. p. 46).

[IKN98] L. Itti, C. Koch et E. Niebur. « A model of saliency-based visual attention for rapid scene analysis ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11 (1998), p. 1254–1259 (cf. p. 44, 45).

[JP87] J. Jones et L. Palmer. « An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex ». Dans : *Journal of Neurophysiology* 58.6 (1987), p. 1233–1258 (cf. p. 30).

[KB01] T. Kadir et M. Brady. « Scale, saliency and image description ». Dans : *International Journal of Computer Vision* 45.2 (2001), p. 83–105 (cf. p. 42).

[Ke04] Y. Ke. « PCA-SIFT : A More Distinctive Representation for Local Image Descriptors ». Dans : *Proccedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 27 June - 2 July 2004, Washington, DC, USA*. T. 2. IEEE Computer Society, 2004. ISBN : 0-7695-2158-4 (cf. p. 47).

[KMY04] D. Kersten, P. Mamassian et A. Yuille. « Object perception as Bayesian inference ». Dans : *Annual Review of Psychology* 55 (2004), p. 271–304 (cf. p. 25).

[KWG94] H. Knutsson, C. Westin et G. Granlund. « Local multiscale frequency and bandwidth estimation ». Dans : *Proceedings of the 1994 International Conference on Image Processing (ICIP 1994), Austin, Texas, USA, November 13-16, 1994*. T. 1. IEEE. IEEE Computer Society, nov. 1994, p. 36–40 (cf. p. 30).

[KZB04] T. Kadir, A. Zisserman et M. Brady. « An affine invariant salient region detector ». Dans : *European Conference on Computer Vision*. 2004, p. 404–416 (cf. p. 42).

[Lin88] R. Linsker. « Self-organization in a perceptual network ». Dans : *Computer Magazine* 21 (1988), p. 105–117 (cf. p. 31).

[Lin96] T. Lindeberg. « Scale-space : A framework for handling image structures at multiple scales ». Dans : *CERN School of Computing*. Egmond aan Zee, The Netherlands, 1996 (cf. p. 34).

[Low04] D. G. Lowe. « Distinctive Image Features from Scale-Invariant Keypoints ». Dans : *International Journal of Computer Vision* 60.2 (2004), p. 91–110 (cf. p. 42, 46, 47).

[Low99] D. G. Lowe. « Object recognition from local scale-invariant features. » Dans : *International Conference on Computer Vision*. T. 2. 1999, p. 1150–1157 (cf. p. 42, 46).

[Mal99] S. Mallat. *A Wavelet Tour of Signal Processing*. 2e éd. Academic Press, 1999 (cf. p. 26).

[Mik+05] K. Mikolajczyk et al. « A Comparison of Affine Region Detectors ». Dans : *International Journal of Computer Vision* 65.1-2 (2005), p. 43–72 (cf. p. 41).

[MPS10] J.-M. Morel, A. B. Petro et C. Sbert. « A PDE Formalization of Retinex Theory ». Dans : *IEEE Transactions on Image Processing* 19.11 (nov. 2010), p. 2825–2837 (cf. p. 29).

[MS02] K. Mikolajczyk et C. Schmid. « An Affine Invariant Interest Point Detector ». Dans : *Lecture Notes in Computer Science* 2350 (2002), p. 128–142 (cf. p. 41).

[MS05] K. Mikolajczyk et C. Schmidt. « A Performance Evaluation of Local Descriptors ». Dans : *IEEE Transactions on Pattern Analysis & Machine Intelligence* 27.10 (2005), p. 1615–1630 (cf. p. 43).

[MZ98] W. Ma et H. Zhang. « Benchmarking of image features for content-based retrieval ». Dans : *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on.* T. 1. Ieee. 1998, p. 253–257 (cf. p. 36).

[OF00] B. A. Olshausen et D. J. Field. « Vision and the Coding of Natural Images ». Dans : *American Scientist* 88 (2000), p. 238–245 (cf. p. 30).

[OF96] B. Olshausen et D. Field. « Emergence of simple-cell receptive field properties by learning a sparse code for natural images ». Dans : *Nature* 381.6583 (1996), p. 607–609 (cf. p. 31, 33).

[OF97] B. Olshausen et D. Field. « Sparse coding with an overcomplete basis set : a strategy employed by V1 ? » Dans : *Vision Research* 37.23 (1997). Vision Res, p. 3311–25 (cf. p. 30).

[OPH96] T. Ojala, M. Pietikäinen et D. Harwood. « A comparative study of texture measures with classification based on featured distributions ». Dans : *Pattern recognition* 29.1 (1996), p. 51–59 (cf. p. 38).

[OT01] A. Oliva et A. Torralba. « Modeling the shape of the scene : A holistic representation of the spatial envelope ». Dans : *International Journal of Computer Vision* 42.3 (2001), p. 145–175 (cf. p. 36).

[PM90] P. Perona et J. Malik. « Scale-space and edge detection using anisotropic diffusion ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (1990), p. 629–639 (cf. p. 34).

[Pra07] W. K. Pratt. *Digital Image Processing*. 4th. Wiley-Interscience, 2007 (cf. p. 26, 28).

[SLC90] P. Schiller, N. Logothetis et E. Charles. « Functions of the color-opponent and broad-band channels of the visual system ». Dans : *Nature* 343 (1990), p. 68–70 (cf. p. 28).

[Sze10] R. Szeliski. *Computer Vision, Algorithms and Applications*. Texts in Computer Science. Springer, 2010 (cf. p. 26, 38).

[TG80] A. Treisman et G. Gelade. « A feature integration theory of attention ». Dans : *Cognitive Psychology* 12 (1980), p. 97–136 (cf. p. 44).

[Tir10] P. Tirilly. « Traitement automatique des langues pour l'indexation d'images ». Thèse de doct. Rennes, France : Université de Rennes 1, juil. 2010 (cf. p. 24, 36, 46).

[TMY78] H. Tamura, S. Mori et T. Yamawaki. « Textural Features Corresponding to Visual Perception ». Dans : *IEEE Transactions on Systems, Man and Cybernetics* 8.6 (juin 1978), p. 460–472 (cf. p. 29).

[TO03] A. Torralba et A. Oliva. « Statistics of natural image categories ». Dans : *Network : Computation in Neural Systems* 14 (2003), p. 391–412 (cf. p. 35).

[Tor+03] A. Torralba et al. « Context-based vision system for place and object recognition ». Dans : *Proceedings of the Nineth IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*. T. 1. IEEE Computer Society, oct. 2003, p. 273–280. ISBN : 0-7695-1950-4 (cf. p. 36).

[Tso92] J. Tsotsos. « On the relative complexity of active v. s. passive visual search ». Dans : *International Journal of computer vision* 7.2 (1992), p. 127–141 (cf. p. 44).

[WCR09] J. Wu, H. Christensen et J. Rehg. « Visual place categorization : Problem, dataset, and algorithm ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE. St. Louis, USA, 2009, p. 4763–4770 (cf. p. 38).

[WR08] J. Wu et J. Rehg. « Where am I : Place instance and category recognition using spatial PACT ». Dans : *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*. T. 2. IEEE. Anchorage : IEEE Computer Society, juin 2008 (cf. p. 38).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 38, 39).

[WWT00] B. Willmore, P. Watters et D. Tolhurst. « A comparison of natural-image-based models of simple-cell coding ». Dans : *Perception* 29.9 (2000), p. 1017–1040 (cf. p. 30).

[Zek93] S. Zeki. *A vision of the brain*. Oxford : Blackwell Scientific Publications, 1993 (cf. p. 26).

Apprentissage artificiel

[Ank+99] M. Ankerst et al. « OPTICS : Ordering Points To Identify the Clustering Structure ». Dans : *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD 1999)*. Sous la dir. d'A. Delis, C. Faloutsos et S. Gandeharizadeh. ACM. Philadelphia, Pennsylvania, USA : ACM Press, juin 1999, p. 49–60 (cf. p. 49).

[BGV92] B. Boser, I. Guyon et V. Vapnik. « A training algorithm for optimal margin classifiers ». Dans : *Fifth Annual Workshop on Computational Learning Theory*. ACM. Pittsburgh, PA, USA, juil. 1992, p. 144–152 (cf. p. 52).

[BT04] G. Bouchard et B. Triggs. « The tradeoff between generative and discriminative classifiers ». Dans : *IASC International Symposium on Computational Statistics (COMPSTAT)*. Citeseer. 2004, p. 721–728 (cf. p. 57).

[CM10] A. Cornuéjols et L. Miclet. *Apprentissage artificiel : concepts et algorithmes*. 2e éd. Paris : Eyrolles, juin 2010, p. 804 (cf. p. 48).

[CV95] C. Cortes et V. Vapnik. « Support-vector networks ». Dans : *Machine learning* 20.3 (1995), p. 273–297 (cf. p. 52).

[DLR77] A. P. Dempster, N. M. Laird et D. B. Rubin. « Maximum likelihood from incomplete data via the EM algorithm ». Dans : *Journal of the Royal Statistical Society, Series B, Methodological* 39 (1977), p. 1–22 (cf. p. 50, 53).

[FD07] B. J. Frey et D. Dueck. « Clustering by Passing Messages Between Data Points ». Dans : *Science* 315 (2007), p. 972–976 (cf. p. 50).

[JH99] T. Jaakkola et D. Haussler. « Exploiting generative models in discriminative classifiers ». Dans : *Advances in neural information processing systems* (1999), p. 487–493 (cf. p. 57).

[Koh90] T. Kohonen. « The Self-Organizing Map ». Dans : *Procceding of the IEEE*. T. 78. 9. 1990, p. 1464–1480 (cf. p. 50).

[NJ02] A. Y. Ng et M. I. Jordan. « On Discriminative vs. Generative classifiers : A comparison of logistic regression and naive Bayes ». Dans : *Advances in Neural Information Processing Systems*. T. 14. MIT Press, 2002 (cf. p. 56, 57).

[Ste56] H. Steinhaus. « Sur la division des corps materiels en parties ». Dans : *Bull. Acad. Polon. Sci* 1 (1956), p. 801–804 (cf. p. 50).

[Vap98] V. Vapnik. *Statistical Learning Theory*. New-York, NY : Wiley, 1998 (cf. p. 51).

[Vap99] V. Vapnik. « An overview of statistical learning theory ». Dans : *IEEE Transactions on Neural Networks* 10.5 (1999), p. 988–999 (cf. p. 51).

II. État de l'art en reconnaissance de scènes

Ce chapitre est consacré aux travaux en reconnaissance de scènes. Ce choix est motivé par le fait que de nombreux travaux en reconnaissance de lieux abordent le problème sous l'angle de la reconnaissance de scènes. De plus, le problème de la reconnaissance de lieux sémantiques possède certaines propriétés que l'exposition préalable des techniques de reconnaissance de scènes nous permettra de mettre en lumière.

La reconnaissance de scènes se place dans le cadre plus général de la reconnaissance visuelle. Cette discipline de la vision par ordinateur fait largement appel aux concepts l'apprentissage artificiel. Nous présentons dans un premier temps les grands concepts et les principales difficultés de la reconnaissance visuelle (section II.1) puis les différentes approches en reconnaissance de scènes (section II.2 et suivantes).

1. Reconnaissance visuelle

La reconnaissance visuelle s'attache principalement à décrire le contenu d'une image et à reconnaître les objets qui la constituent. C'est sans doute la tâche la plus difficile en vision par ordinateur. Ainsi, malgré les importants progrès réalisés ces dernières années, Szeliski [Sze10, p612] considère que les meilleurs systèmes sont moins bons qu'un enfant de deux ans.

Bien que nous nous intéressions dans cette partie à la reconnaissance de scènes, il est difficile de ne pas aborder les tâches liées à la reconnaissance visuelle des objets. En effet, la plupart des travaux que nous présenterons ont été développés pour la reconnaissance ou la catégorisation d'objets dans une image. Les études en reconnaissance d'objets ont donc une forte influence sur le domaine de la reconnaissance de scènes.

Après avoir précisé l'emploi que nous ferons de quelques termes, les sous-sections II.1.2 et II.1.3 seront consacrées aux motivations et aux difficultés de la reconnaissance d'objets et de scènes respectivement. La Section II.1.4 présentera un aperçu des différentes méthodes qui seront développées dans les section II.2 et suivantes.

1.1. Introduction

La reconnaissance visuelle renvoie à des tâches et des concepts dont la nomenclature est parfois équivoque. Pour clarifier notre propos, nous définissons l'emploi que nous faisons des termes de base.

1.1.1. Objets, scènes et classes

Le terme « objet » désigne un objet particulier que l'on cherche à reconnaître. Cela inclut les objets habituels (la tasse de Sylvain, l'écran de l'ordinateur de Cédric, *etc.*) mais aussi les visages (le visage de Ignace), les piétons, *etc.*

Le terme « scène » est parfois utilisé au-sens de pré-image (voir section I.2). Cependant, dans notre travail, le mot scène désigne un sous-ensemble des images naturelles dont on exclut les images portant sur une partie d'un environnement ou un objet seul (voir Figure II.1). On appelle parfois ce sous-ensemble les scènes naturelles mais il contient aussi des scènes de villes, d'intérieur, *etc.* Aussi, nous réserverons le terme de « scène naturelle » aux scènes non-fabriquées par l'homme. Une scène correspond à une partie du champ visuel perçu par un être humain dans les conditions naturelles de vision. D'après Henderson [Hen05b], une scène est une partie cohérente, nommable et à échelle humaine d'un environnement réel. La notion de distance entre l'observateur et la pré-image est donc cruciale. Dans la plupart des images d'objets, celle-ci est inférieure à 1 m ou 2 m. Pour les scènes, Oliva et Torralba [OT01] considèrent qu'elle est supérieure à 5 m. Une scène est donc un espace dans lequel on peut se déplacer. Les éléments de la scène peuvent être répartis entre un avant-plan et un arrière-plan.

Le terme « catégorie » désigne un ensemble d'entités réunies en vertu d'une ou plusieurs caractéristiques communes. Ces caractéristiques peuvent relever de critères sensoriels, fonctionnels ou sémantiques. On peut parler de catégories d'objets (par exemple, la catégorie des tasses) ou de catégories de scènes (par exemple les scènes de montagne). Les catégories sont en général hiérarchisées : une catégorie peut faire partie d'une catégorie d'un niveau supérieur (par exemple les scènes de montagne appartiennent à la catégorie des scènes naturelles). Une catégorie de niveau supérieur est appelée catégorie sur-ordonnée.

Enfin, nous réserverons le terme « classe » aux sorties possibles d'un algorithme de classification.

1.1.2. Tâches

La reconnaissance d'instances ou identification (*instance recognition* ou simplement *recognition*) désigne le processus par lequel une même entité est reconnue dans une image, indépendamment des variations de prise de vues. En ce qui concerne les objets, cela signifie reconnaître la tasse de Sylvain quel que soit l'angle de vue, la luminosité, *etc.* En ce qui

(a) Vues rapprochées d'un objet

(b) Scènes

FIGURE II.1.: Exemples d'images naturelles qui ne sont pas des scènes (a) et de scènes (b). Les images (a) sont issues de la base de données Caltech 256 (voir [GHP07]) pour la catégorisation d'objets. Les images (b) sont issues de la base décrite dans [OT01] pour la reconnaissance de scènes.

concerne les scènes, la tâche consiste à reconnaître la même scène sous un angle différent (ex : reconnaître les scènes incluant le Mont-Blanc).

De même, la catégorisation ou reconnaissance de catégorie (*category-level recognition* ou *generic recognition*) désigne le processus par lequel une entité est reconnue comme appartenant à une catégorie donnée. Par exemple, la tasse de Sylvain sera reconnue comme appartenant à la catégorie des tasses en général. Pour les scènes, une image contenant le Mont-Blanc sera catégorisée comme une scène de montagne. Les catégories sont, en général, prédéfinies (la catégorisation ne désigne donc pas les critères de création des catégories).

Nous utiliserons le terme « reconnaissance » pour parler indistinctement de catégorisation et d'identification. On utilise parfois le terme « classification » dans ce sens.

La plupart des méthodes d'identification ou de catégorisation travaillent sur une sous-fenêtre. Ainsi, pour traiter une image entière, on devrait appliquer cet algorithme sur toutes les sous-fenêtres, ce qui est lent. La tâche de détection ou localisation consiste à donner les positions probables d'une catégorie d'objets dans une image sans identifier ces objets (ex : la position probable des visages, des piétons, *etc.*). Une fois ces fenêtres localisées, on peut utiliser un algorithme de reconnaissance d'instances ou de catégories.

La tâche de description de scènes consiste en l'énumération du contenu d'une scène, ce qui implique, en général, la détection et la catégorisation des objets présents dans la scène.

Du point de vue de l'apprentissage, les tâches de détection, de reconnaissance d'instances et de catégorisation sont des tâches de classification souvent abordée de manière supervisée (voir Figure II.2) :

- Dans la tâche de détection, chaque pixel ou sous-fenêtre est classé comme appartenant à une catégorie ou au fond.
- Dans la tâche de reconnaissance d'instances, chaque image est classée comme contenant un objet ou non. En général, l'algorithme fournit d'autres informations comme la position de l'objet dans l'image (boîte englobante ou éventuellement segmentation précise).
- Dans la tâche de catégorisation d'objets, chaque image est classée comme contenant une catégorie ou non.

Dans ces différents cas, les sorties de l'algorithme d'apprentissage (les classes) sont différentes.

Un domaine voisin, que nous avons déjà mentionné, est celui de l'indexation de larges bases de données d'images (*Content Based Image Retrieval (CBIR)*). Le but est de fournir l'équivalent pour les images des systèmes de recherche d'information textuelle (*e.g.* les moteurs de recherche sur Internet) [1]. On dispose donc d'une base d'images qui jouent le rôle des documents. Lorsque l'utilisateur présente une image requête, on doit lui retourner un certain nombre de documents « similaires » issus de la base. Suivant la tâche, on s'intéresse soit aux images de scènes identiques soit aux images de la même catégorie sémantique. Ici aussi, la

1. Au moment de la rédaction de cette thèse, Google a lancé pour le grand public son service de recherche à partir d'une photo. Bien que des sites spécialisés proposaient déjà de tels services, c'est, à notre connaissance, le premier moteur de recherche grand public à proposer un tel système.

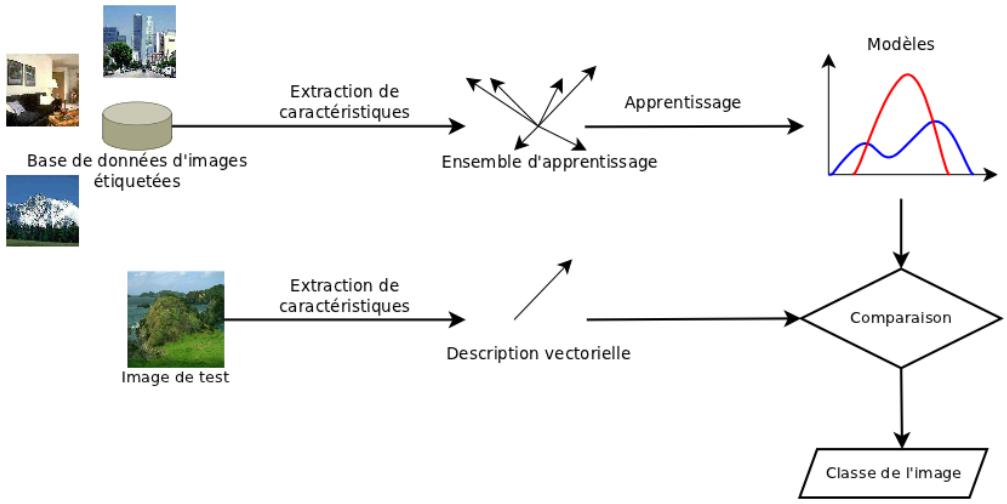


FIGURE II.2.: Schéma de principe de la reconnaissance visuelle.

question de la description des images se pose. Ce domaine a vu fleurir de nombreux travaux ces dernières années, c'est pourquoi nous en parlerons.

Ces tâches entretiennent un rapport étroit et la plupart des techniques développées dans un cadre peuvent être ré-utilisées dans un autre. La section suivante est dédiée à la présentation de la reconnaissance d'objets.

1.2. Reconnaissance d'objets

Un être humain reconnaît spontanément et sans effort les objets de son environnement. On estime qu'un être humain est capable de reconnaître plus de 10 000 catégories d'objets ([Bie95]) et un nombre probablement plus grand d'objets particuliers.

1.2.1. Intérêt pour la robotique cognitive

Pour un robot mobile, la reconnaissance d'objets peut servir à repérer des amers de navigation. La difficulté est qu'il est souvent plongé dans un environnement complexe. Un autre intérêt de la reconnaissance d'objets est la manipulation : cette tâche nécessite en général un modèle 3D d'un objet spécifique ainsi que sa localisation précise mais est facilitée par le fait que l'objet est perçu de près, généralement isolé, posé sur un fond uniforme (voir [Cot07]). Enfin, dans le cadre de la robotique mobile de service, la reconnaissance d'objets est indispensable à la communication homme-machine.

1.2.2. Difficultés de la reconnaissance d'objets

Les difficultés de la reconnaissance visuelle viennent de ce que l'on cherche à rapporter à une même catégorie ou à une même instance des entrées visuelles parfois très hétérogènes. L'identification est difficile car les images d'un même objet peuvent être très différentes l'une de l'autre. On parlera de variabilité intra-objet. Dans les tâches de catégorisation viennent s'ajouter des différences visuelles parfois très importantes entre deux éléments d'une même catégorie. On parlera de variabilité intra-catégorie.

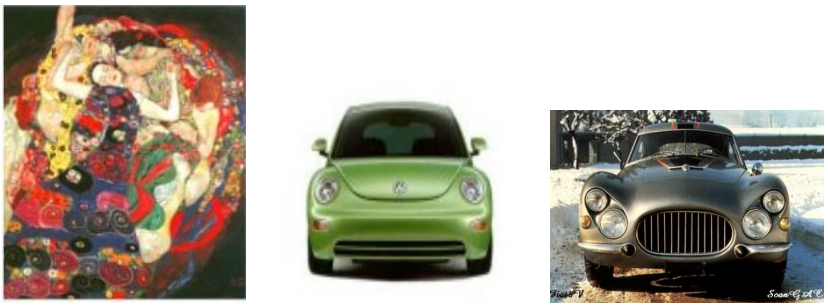
En reconnaissance d'objets, FERGUS [Fer05] distingue six sources de variabilité (voir Figure II.3) :

1. Les changements de point de vue (voir figure II.3a) : du point de vue de l'observateur, les objets peuvent subir toute une série de transformations : rotation dans le plan, translation, changement d'échelle, *etc.* Chacune de ces transformations modifie l'apparence de l'objet de façon plus ou moins importante. En général les transformations en dehors du plan, par exemple la rotation en profondeur, posent le plus de difficultés car elles offrent à la vue des faces différentes du même objet.
2. Les différences d'illumination (voir figure II.3b) : les changements d'illumination modifient la valeur des pixels de l'image. Un changement d'intensité peut décaler ou modifier l'échelle des valeurs. Une modification de la position de la source lumineuse entraîne généralement des problèmes plus complexes, les ombres portées conduisant à des modifications non linéaires des valeurs de pixels.
3. L'occultation (voir figure II.3c) : certaines parties d'un objet peuvent être occultées par un autre objet ou par une autre partie de l'objet lui-même.
4. Les déformations (voir figure II.3d) : certains objets sont déformables ce qui conduit à des modifications dans la taille et la forme des parties les constituants, ainsi que dans les relations spatiales entre ces parties.
5. La présence d'un arrière-plan complexe (voir figure II.3e) : la séparation figure/fond est un problème difficile lorsque l'arrière-plan est complexe. Le problème est principalement de segmenter l'image de sorte à isoler l'objet d'intérêt des autres, considérés comme distracteurs.
6. La variabilité intra-catégorie (voir figure II.3f) : les instances d'une même catégorie peuvent avoir des apparences visuelles très différentes.

La reconnaissance d'instances est un sujet assez bien maîtrisé (des applications commerciales sont déjà disponibles) mais la catégorisation est encore difficile, en particulier quand le nombre de catégories est grand. Traditionnellement, la catégorisation s'est concentrée sur des catégories relativement simples (par exemple les chiffres, les voitures, *etc.*). La plupart des bases de données contiennent quelques dizaines de catégories (par exemple, le défi PASCAL 2008 définit 20 catégories).

(a) Variations de points de vue (b) variations d'illumination

(c) Occlusion (d) Déformation



(e) Arrière-plan chargé (f) Variation intra-catégorie

FIGURE II.3.: Sources de variations rencontrées en reconnaissance d'objets. Adapté de [Fer05 ; FFT07].

(a) Catégorie perceptuelle (b) Catégorie fonctionnelle

FIGURE II.4.: Exemple de catégorie perceptuelle ((a)) et de catégorie définie par des critères fonctionnels ((b)). D'après [Fer05].

Un problème en catégorisation est qu'une catégorie peut réunir des objets en fonction de critères fonctionnels ou sémantiques, étrangers au domaine visuel (voir Figure II.4). Par exemple, la catégorie « chaise », définie par un critère fonctionnel (« objet sur lequel on peut s'asseoir »), montre une grande variabilité visuelle. La plupart des études se concentrent sur les catégories « visuelles » c'est-a-dire constituées selon des critères perceptuels (par exemple les voitures).

La détection est souvent employée pour faciliter l'identification en restreignant les zones à traiter. Cependant, un algorithme de détection doit repérer toutes les instances d'une catégorie d'objets. Ce problème devient compliqué si l'on cherche à localiser précisément plusieurs catégories d'objets ou que le nombre d'objets par catégorie n'est pas connu (par exemple détecter les piétons, les voitures et les immeubles dans une rue).

De plus, les êtres humains sont capables de reconnaître et de décrire des objets qu'ils n'ont jamais vus en faisant appel à des attributs perceptuels (l'objet est brillant, l'objet est texturé, *etc.*), aux parties qui composent l'objet (l'objet a des yeux, une queue, *etc.*) ou encore à des informations de catégorie (c'est un véhicule, c'est un animal, *etc.*). Les propriétés servant à décrire un objets sont appelées des propriétés utiles. De nouvelles approches en reconnaissance (voir [Far+09]) se basent sur ce mode de descriptions des objets. Un objet n'est plus vu comme appartenant à une certain classe mais comme une collection d'attributs. Ces approches, bien qu'elles soient à leur début, permettent d'envisager une reconnaissance plus souple.

1.3. Reconnaissance de scènes

Le but de la reconnaissance de scènes est de reconnaître le contenu d'une scène visuelle dans son ensemble et de la classer parmi un ensemble de catégories sémantiques. Par exemple

une image pourra être étiquetée comme « le Mont-Blanc », ou à un niveau plus général comme une « chaîne de Montagnes ».

On distingue traditionnellement les scènes naturelles (les paysages de montagne, de forêts, de plage) des scènes fabriquées par l'homme, parmi lesquelles, on distingue les scènes d'extérieur (ville, route) et les scènes d'intérieur (chambre, salon, *etc.*).

Les catégories pertinentes et les hiérarchies sont difficiles à établir. Aussi, de nombreuses études psychologiques ont été conduites afin de déterminer la manière dont les êtres humains classent les scènes. Dans les études expérimentales (voir [TH83]), les catégories sur-ordonnées (*i.e.* les catégories les plus générales) sont d'un côté les scènes d'intérieur et de l'autre les scènes d'extérieur. Les scènes d'extérieur se composent des catégories « ville », « parc », « plage » et « montagne ». Les scènes d'intérieur se répartissent entre les catégories « restaurant », « magasin », « rue » et « maison ».

Comme nous l'avons dit (voir Section I.2.1.4), il est couramment admis que le système visuel humain a été façonné pour permettre d'extraire de l'information des scènes naturelles. Les expériences de Potter (voir [Pot75]) montrent que l'on reconnaît une scène en moins de 200 ms, ce qui est du même ordre que le temps nécessaire pour reconnaître les objets (voir aussi [RJF05]). De même, Biederman, Mezzanotte et Rabinowitz [BMR82] montrent qu'un objet est reconnu plus rapidement dans un contexte familier que dans un contexte improbable. Enfin, les travaux de Fei-Fei et al. (voir [Fei+05]) montrent que la reconnaissance de scènes nécessite peu de charge attentionnelle. En fait, tous les stimuli habituels et signifiants sont reconnus avec peu d'attention spatiale (ils sont reconnus « en un coup d'œil ») : on reconnaît facilement les T et les L dans leurs positions habituelles mais si ils sont tournés, la tâche devient beaucoup plus dure. Cela montre l'importance du contexte et que celui-ci doit être traité avant la reconnaissance d'objets donc de manière holistique (c'est-à-dire sans le décomposer en éléments). D'un point de vue neurologique, une région du gyrus parahippocampique (appelée *parahippocampal place area*) répond sélectivement aux scènes (et non aux objets ou aux visages). De plus, la réponse est la même qu'elles contiennent ou non des objets (voir [EK98 ; Eps05]). Cela tend à démonter l'existence de mécanismes spécifiques pour l'analyse des scènes.

Comme nous l'avons dit, la reconnaissance d'objets travaille en général avec des vues rapprochées des objets. À l'opposé, les scènes sont des images dans lesquelles la distance entre l'observateur et ce qui est observé est grande. Contrairement au cas de l'observation rapprochée où un objet peut être observé sous n'importe quel angle, les contraintes imposées par la taille des scènes et donc la distance d'observation impose des contraintes sur le point de vue. Par exemple, on ne peut pas observer une montagne de près : les montagnes sont en général perçue au fond de l'image, avec un distance typique de plusieurs centaines ou milliers de mètres et le ciel au-dessus (voir figure II.1b). Les autres catégories de scènes imposent aussi des contraintes sur le point de vue et la disposition des différents éléments (il est difficile d'observer une plage sans voir la mer et le ciel en fond). Torralba et Oliva [TO03] ont analysé les propriété statistiques des scènes (rappelons que ce travail est à l'origine

du descripteur GIST vu précédemment). Les principales conclusions sont :

- Il existe des directions privilégiées : le spectre de puissance moyen des scènes n'est pas isotrope. De plus, les directions préférentielles sont caractéristiques de la catégorie (voir figure II.5a).
- Les statistiques dépendent de la distance d'observation : une même catégorie n'a pas les mêmes propriétés à différentes distances (voir figure II.5b).
- Les statistiques ne sont pas stationnaires aux grandes échelles (voir figure II.5c). Cela traduit le fait que, observées de loin, les scènes possèdent une certaine structure du fait des limitations du point de vue.

Ces propriétés justifient les méthodes vues dans la Section I.2.2 permettant d'inclure des information géométriques dans les descripteurs globaux.

1.3.1. Intérêt pour la robotique cognitive

Comme nous l'avons déjà signalé, Shakey (voir Section I.1.3.1 page 13) utilisait un système d'analyse de scènes pour se localiser. L'analyse et la description de scènes est un des domaines clés dans l'optique de la robotique personnelle.

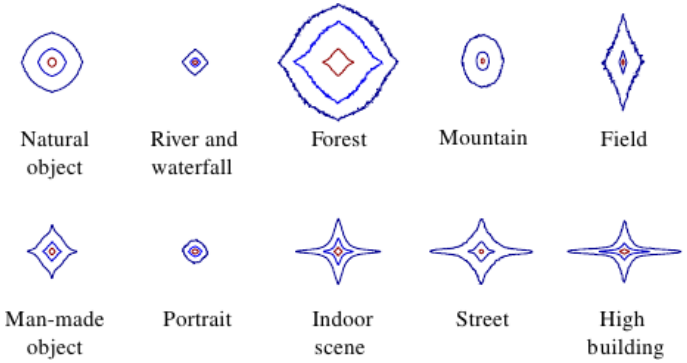
Dans le contexte de la vision par ordinateur, la reconnaissance de scènes est souvent appelée reconnaissance de contexte. En effet, la connaissance de la catégorie sémantique de la scène permet de donner des indices sur la présence et la localisation de certains objets (voir [Tor03 ; OT07]). La première utilité de la reconnaissance de scènes est donc d'améliorer les capacités à détecter et reconnaître les objets.

La reconnaissance de scènes entretient un rapport étroit avec la reconnaissance de lieux que nous détaillerons dans le chapitre III. Nous verrons cependant que la reconnaissance de lieux est un problème spécifique, lié au caractère situé de la plate-forme robotique. La différence principale tient au fait qu'une scène n'est définie que comme une image et donc sa reconnaissance fait appel uniquement aux pixels de l'image. Un lieu est une entité définie dans l'espace réel et sa reconnaissance fait donc appel à d'autres informations. De plus, la notion de contexte est plus importante dans la reconnaissance de lieux sémantiques.

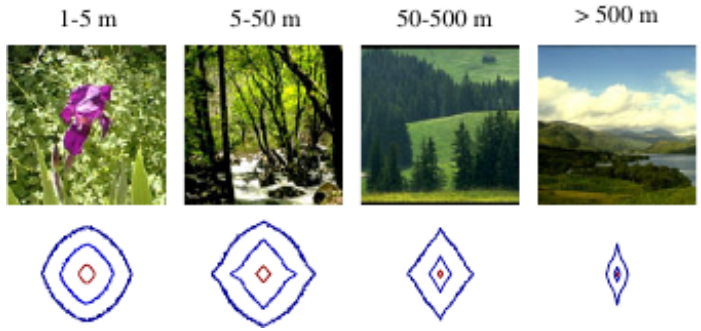
1.3.2. Difficultés

Un certain nombre de difficultés de la reconnaissance d'objets se retrouvent pour la reconnaissance de scènes : les changements liés à l'illumination et la variabilité intra-catégorie se posent avec la même acuité. De même, le problème de la déformation se pose lui aussi : le fait que certains composants de la scène soient déplacés ne change évidemment pas son contenu sémantique.

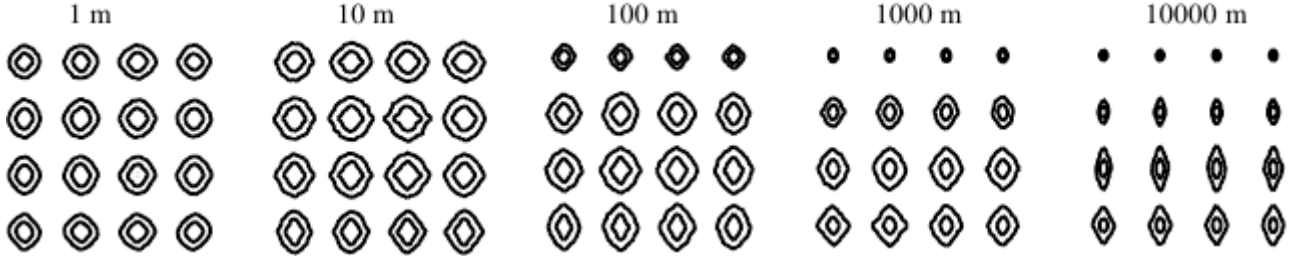
Les problèmes liés aux changements de point de vue se posent aussi mais de manière différente : le changement de point de vue est en général constitutif de la définition d'une



(a) Spectre de puissance moyen des scènes naturelles pour différentes catégories. La figure montre les courbes de niveau représentant 60%, 80% et 90% de l'énergie totale.

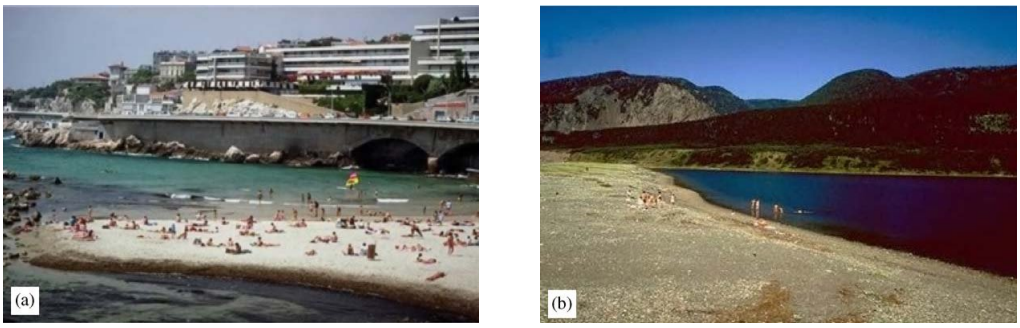


(b) Influence de la distance d'observation sur le spectre des scènes de forêt. La figure montre les courbes de niveau représentant 60%, 80% et 90% de l'énergie totale.



(c) Non-stationnarité du spectre des scènes naturelles. Dans ce cas, le spectre est calculé dans chacune des 4×4 sous-fenêtres.

FIGURE II.5.: Propriétés statistiques des scènes naturelles. Tiré de [TO03].



(a) plage et ville (b) plage et montagne

FIGURE II.6.: Illustration du problème de multi-classification : les images présentées ont été classées (par des sujets humains) comme appartenant aux deux catégories indiquées. Extrait de [Bou+04].

scène.

En revanche, les problèmes d'occultation et d'arrière-plan ne se posent pas. En effet, une scène visuelle n'a pas d'autre extension spatiale que celle temporaire imposée par les limites de l'image. Une scène visuelle n'est jamais que ce que l'on voit, et la description que l'on peut en faire. Par conséquent, il n'est pas possible de parler d'occultation ou d'arrière-plan d'une scène. En revanche, si l'on cherche à décrire la scène en reconnaissant les objets qu'elle contient, alors ces problèmes vont se poser.

BOUTELL et al. [Bou+04] font remarquer qu'il est souvent possible de catégoriser une image de plusieurs façons non-exclusives (voir Figure II.6). Ce problème, appelé multi-classification, est une difficulté particulière de la catégorisation de scènes. Dans le problème de classification standard, les erreurs sont dues au recoupement des classes alors que dans un problème de multi-classification, les classes peuvent se recouvrir par définition : les images sont partie intégrante de plusieurs classes. Ce problème est aussi distinct (mais proche) de la classification floue dans laquelle une image appartient à certaines classes à différents degrés.

Pour revenir sur le problème de l'occultation, si l'on imagine un paysage de montagne masqué au premier plan par un chalet, le problème de savoir s'il faut rapporter cette scène à la classe « montagne » ou « bâtiment » est, en définitive, un problème de multi-catégorisation.

Le reste du chapitre présente un état de l'art en reconnaissance des scènes.

1.4. Vue d'ensemble des techniques utilisées en reconnaissance de scènes

Historiquement, les premières approches en vision postulent que la perception d'une scène est la résultante de la perception des objets qui la composent (ses composantes individuelles). L'accès au sens d'une scène est la dernière étape du processus d'analyse. Un exemple classique est la théorie de Marr (voir [Mar82] et Section II.4.2) : dans cette approche, les objets sont d'abord isolés de l'environnement (arrière fond) pour être reconnus. D'autres approches considèrent au contraire que l'on peut reconnaître une scène de manière holistique, sans reconnaître les objets.

Une autre distinction importante concerne la nécessité des informations locales, c'est-à-dire l'importance de la disposition précise de la scène. Les expériences de Biederman (voir [Bie72]) ont montré qu'une scène dont on a mélangé les parties est reconnue moins rapidement qu'une scène cohérente (voir Figure II.7). Le terme anglais utilisé pour décrire cette représentation globale est appelée *gist*[2]. Le descripteur GIST (voir page 36) est une description quantitative de cette représentation psychologique. De nombreux travaux ont étudié les propriétés de cette représentation (voir [Oli05 ; OT06] pour une revue) :

- Les fréquences basses sont, en général, suffisantes pour la reconnaissance de la catégorie sémantique d'une scène. Cela ne doit pas être interprété comme une propriété du système visuel mais montre plutôt l'existence d'un mécanisme de sélection de l'échelle dépendant de la tâche. Pour la plupart des tâches de catégorisation de scènes, les basses fréquences portent l'information intéressante et ce sont donc celles que le système visuel utilise. Il est cependant possible de favoriser au contraire les hautes fréquences (voir [OS97]).
- La distribution grossière des couleurs peut influer sur la catégorisation pour les scènes dans lesquelles la couleur est importante (typiquement les scènes naturelles).

Plus récemment, Vogel et al. [Vog+06] ont étudié l'impact du mélange des parties (qui détruit l'information globale mais conserve l'information locale) et de l'ajout de flous (qui détruit l'information locale mais conserve l'information globale). En moyenne, l'effet du mélange est aussi important que l'effet du flou (voir Figure II.7) : le taux de reconnaissance passe d'environ 90% à 70% (les images sont présentées pendant 4 s). Cependant, l'amplitude de l'effet de chaque modalité et leur importance relative dépend de la catégorie. Par exemple pour la catégorie « rivière/lac » l'effet du mélange est plus important que le flou. Pour les côtes, le flou perturbe plus la catégorisation que le mélange. Enfin pour les forêts, il n'y a quasiment pas d'effet. Ces résultats montrent que les deux types d'information sont

2. Le lecteur passionné d'étymologie notera que le mot *gist* vient du vieux français gésir (qu'on retrouve dans l'expression « ci-gît »). L'usage originel en anglais vient du domaine juridique et désigne la situation où une loi s'applique : "cest action gist" signifie "ce cas est défendable". Le sens moderne en anglais (qu'on retrouve parfois dans le verbe gésir) correspond à "l'essence, l'idée principale de quelque chose" et est attesté dès 1823. Une traduction d'inspiration heideggerienne serait « le ci-gît d'une scène ». Sources : http://www.etymonline.com/index.php?term=gist et http://fr.wiktionary.org/wiki/g%C3%A9sir.

nécessaires et intégrés pour la catégorisation de scènes.

Nous nous inspirons du plan de [BMM07] en distinguant :

1. les approches basées sur les caractéristiques bas-niveau (voir section II.2)
2. les approches basées sur la quantification de propriétés locales (voir section II.3)
3. les approches basées sur une description sémantique de l'image (voir section II.4)

Notons qu'il est difficile de catégoriser univoquement les techniques. Ainsi la classification présentée n'est pas la seule possible et sera parfois redondante.

2. Approches basées sur les caractéristiques bas-niveau

Les approches basées sur les caractéristiques bas-niveau considèrent que les scènes peuvent être classées à partir des informations contenues dans l'image (luminosité, couleur et textures). Pour cela, on calcule une description de l'image qui est utilisée par un classificateur. Comme pour la caractérisation des images, on peut distinguer les approches globales (voir Section II.2.1) et locales (voir Section II.2.2). Enfin, les techniques statistiques ont aussi été appliquées dans ce cadre (voir Section II.2.3).

2.1. Approches globales

Dans les approches globales, l'image est caractérisée globalement comme nous l'avons vu dans la Section I.2.2. Cette représentation est utilisée pour classer l'image. Nous distinguons les méthodes avec et sans géométrie.

2.1.1. Sans géométrie

Dans ce cas, un seul vecteur est construit qui représente toute l'image. Différentes caractéristiques peuvent être utilisées : histogramme d'orientation ([SB91]), couleur ([LL03 ; MZ98 ; BBV02]), textures ([HR04]). Un certain nombres d'approches utilisent plusieurs caractéristiques à la fois ([SSN05 ; VJZ98 ; Vai+01]).

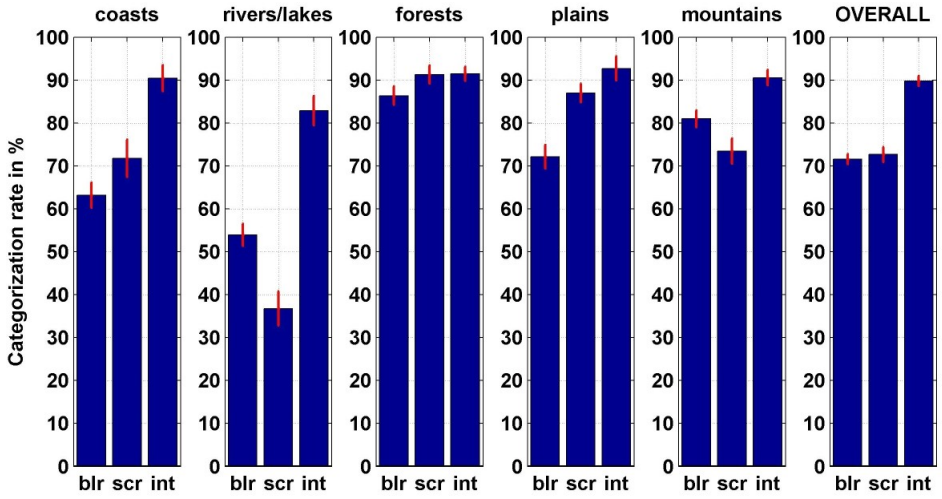
Parmi les classificateurs employés, on trouve les Support Vector Machines (SVM), les classificateurs bayésiens naïfs et les k plus proches voisins (k-PPV).

Les travaux de Vailaya (voir [VJZ98 ; Vai+01]) sont représentatifs de cette approche. Dans un premier temps, le système différencie les scènes d'extérieur et les scènes d'intérieur en utilisant un histogramme des couleurs. Dans un deuxième temps, les scènes de ville sont distinguées des scènes de montagne à l'aide d'un histogramme d'orientation (voir Figure II.8). Le classificateur utilisé est génératif : les densités conditionnelles sont modélisées par un mélange de Gaussiennes dont les centres sont obtenus par l'utilisation d'un algorithme

(a) Image originale

(b) Image mélangée (*scrambled*) : seule l'information locale est conservée

(c) Image floutée (*blurred*) : l'information locale est masquée mais l'information globale est conservée



(d) Taux de catégorisation correcte en fonction de la condition (int = intacte, blr = *blurred*, scr = *scrambled*)

FIGURE II.7.: Importance de l'information globale et locale pour la catégorisation de scènes. Tiré de [Vog+06]

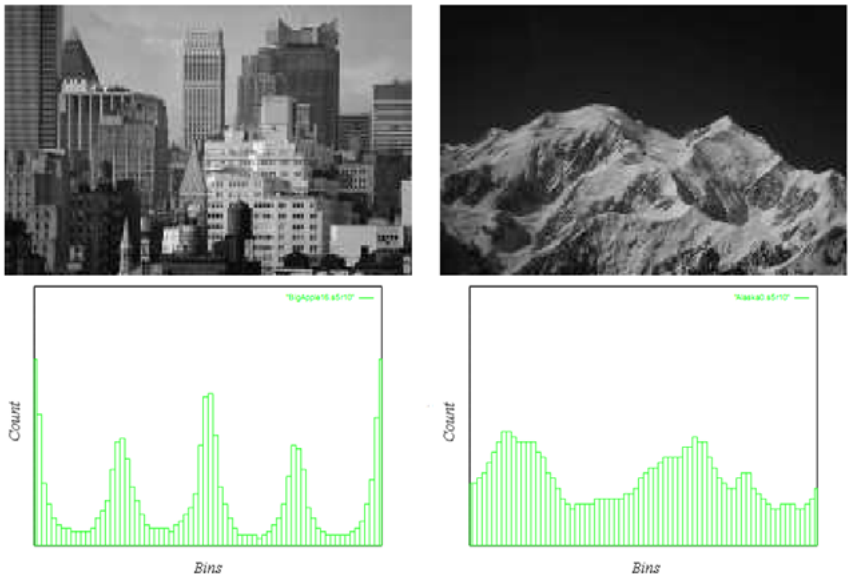


FIGURE II.8.: Utilisation de l'histogramme d'orientation pour discriminer les scènes d'intérieur et les scènes d'extérieur. Les scènes d'intérieur sont caractérisées par la dominance des orientations verticale et horizontale, ce qui produit des histogrammes très irréguliers. Les scènes naturelles produisent des histogrammes plus plats. Source : [VJZ98]

de quantification vectorielle. Cette méthode entretient des liens étroits avec les méthodes Bag-of-Words que nous présentons dans la section II.3.

2.1.2. Avec géométrie grossière

Rappelons que la géométrie grossière (voir Section I.2.2) consiste à subdiviser l'image selon une grille et à calculer un descripteur sur chaque région. On peut distinguer d'un côté les méthodes qui classifient chaque bloc indépendamment puis combinent ces décisions pour classer l'image (voir [SSL02 ; SP98]) et d'un autre côté les méthodes qui classent directement toute l'image (voir [GP94]).

Les indices utilisés sont les mêmes que dans le premier cas : orientation ([GP94]), couleur et texture ([SSL02]). SZUMMER et PICARD [SP98] comparent différents codages (couleur, textures et les coefficients de la transformée en cosinus discrète) et leurs combinaisons. La combinaison de plusieurs codages augmente les performances mais les performances sont difficiles à prévoir (la combinaison la plus avantageuse ne fait pas appel à l'orientation). Ils trouvent aussi qu'un classificateur k-PPV donne de meilleurs résultats qu'un réseau de neurones. Signalons que la norme MPEG-7 utilise un codage basé sur les histogrammes

d'orientation (voir [Man+01]).

Douze et al. [Dou+09] évalue GIST dans une tâche de recherche d'image. Les résultats le placent au niveau des meilleures méthodes basées sur les mots visuels (voir section II.3).

Wu [Wu09, chap. 3] compare CENTRIST et GIST dans une tâche de reconnaissance de scènes, la classification étant obtenue avec des SVM. GIST montre de bonnes performances pour la classification de scènes d'extérieur mais les performances chutent pour les scènes d'intérieur. L'explication de l'auteur est que les différentes scènes d'intérieur ont des orientations très proches que GIST ne permet pas de discriminer. Les performances de CENTRIST sont intéressantes mais doivent être confirmées par d'autres études.

2.2. Approches locales

Les approches locales consistent à extraire des points d'intérêt dans l'image (voir page 40) et les descripteurs associés. Ces approches sont très populaires en reconnaissance d'objets, notamment depuis les travaux de Lowe (voir [Low99 ; Low04]) et ont aussi été appliquées à la recherche d'images similaires.

Le principe de base est d'extraire et de caractériser des points d'intérêt dans l'image et de comparer ces descripteurs avec les images de référence. L'utilisation de ces techniques nécessite des structures de données adaptées au stockage de nombreux vecteurs de plus de 100 dimensions et d'algorithmes de recherche approximative de plus proches voisins en grande dimension.

Dans le cas de la reconnaissance d'objets, les images de référence sont des images de l'objet sans fond, ce qui permet d'avoir un modèle géométrique de l'objet (les positions, l'échelle et l'orientation des points d'intérêt sont stockées). Dans une tâche de reconnaissance d'objets, l'objet original peut être placé dans un arrière-plan complexe et partiellement occulté. Il est donc nécessaire de trouver sa position, son échelle et son orientation (ce qui est possible car on stocke les coordonnées des points d'intérêt). Cette étape, qu'on appelle l'alignement ou le recalage du modèle, procède en deux temps : dans un premier temps, on compare les descripteurs trouvés dans l'image de test aux descripteurs des images de la base de données ; quand un nombre suffisant de correspondances a été trouvé, le système vérifie si les positions des points sont compatibles avec le modèle géométrique de l'objet (cette étape est appelée *match verification*). Ces techniques permettent une détection robuste des objets à plusieurs échelles, sous plusieurs orientations, partiellement occultés et en présence d'un arrière-plan complexe. L'utilisation des contraintes géométriques est cruciale.

Ces approches ont été utilisées pour la recherche de scènes identiques (voir [SM97 ; LJJ08]). Le nombre de descripteurs n'étant pas le même pour chaque image (sauf dans le cas particulier d'un échantillonnage dense), il est impossible de comparer directement les descriptions. On utilise en général un système de vote : l'image la plus ressemblante est celle qui a le plus de descripteurs proches (au sens de la distance entre vecteurs) de l'image de test. La vérification géométrique est plus difficile car la géométrie des scènes est variable. Pour ces raisons, il est

difficile d'utiliser les méthodes bas-niveau locales pour des tâches de catégorisation de scènes car les différentes instances d'une même catégorie sont très variables.

2.3. Approches statistiques

Une autre approche possible consiste à analyser les statistiques des scènes naturelles pour extraire les descriptions pertinentes, en utilisant l'Analyse en Composantes Principales (ACP) ou l'Analyse en Composantes Indépendantes (ACI) (voir [LBP99]). Ces approches sont très populaires en reconnaissance d'objets, notamment depuis les travaux de Turk et Pentland (voir [TP91]) sur les « visages propres ».

Les travaux de Le Borgne (voir [Le 04 ; LBGO07]), basés sur l'ACI, sont représentatifs de cette approche. Partant d'une collection d'images, on extrait des imagettes de taille 32×32 et on effectue une réduction de la dimension via une ACP. Pour compenser les effets de bord, on applique une fenêtre de Hamming (cette étape est appelée apodisation). On applique ensuite l'algorithme fastICA (voir [HCO99]) afin d'extraire les composantes indépendantes de ces imagettes. Les filtres obtenus sont approximés par des filtres de Gabor (voir [Hat92 ; HS98]).

L'un des avantages de l'ACI par rapport à l'ACP est que les filtres obtenus sont adaptés aux catégories alors que l'ACP produit des filtres non-spécifiques (voir Figure II.9).

La représentation de l'image se base sur l'activité des filtres. Il existe plusieurs manières de faire :

Densité de probabilité de réponse des filtres : dans cette approche, on modélise la densité de probabilité de l'énergie de chaque filtre sur l'ensemble des images d'une catégorie. Différentes approches sont possibles : paramétriques (gaussienne, demi-gaussienne, famille exponentielle), non-paramétriques (estimation par noyaux, estimation log-spline ou histogramme logarithmique). La probabilité jointe (pour la classe c) est le produit des probabilités marginales (car les composantes sont indépendantes). Pour une image de test, on calcule la réponse des N filtres. La distance entre l'image de test et la classe c est calculée avec la divergence de Kullback-Leibler entre la densité observée et la densité de chaque classe. La classe la plus proche correspond à la classe choisie par la règle de Bayes (voir section I.3).

Histogramme d'activité des filtres : pour chaque pixel, on note le numéro du filtre le plus actif puis on dresse l'histogramme de l'activité des filtres [3]. Une variante utilise la réponse sur une fenêtre, ce qui conduit à des histogrammes localisés. La classification se fait ensuite en utilisant des SVM. L'approche par activité de filtres est plus simple et plus robuste que l'approche par modélisation de la densité.

3. Cette technique est parfois appelée *bag-of-activity* en raison de l'absence d'information spatiale comme dans les approches sac-de-mots (voir section II.3).

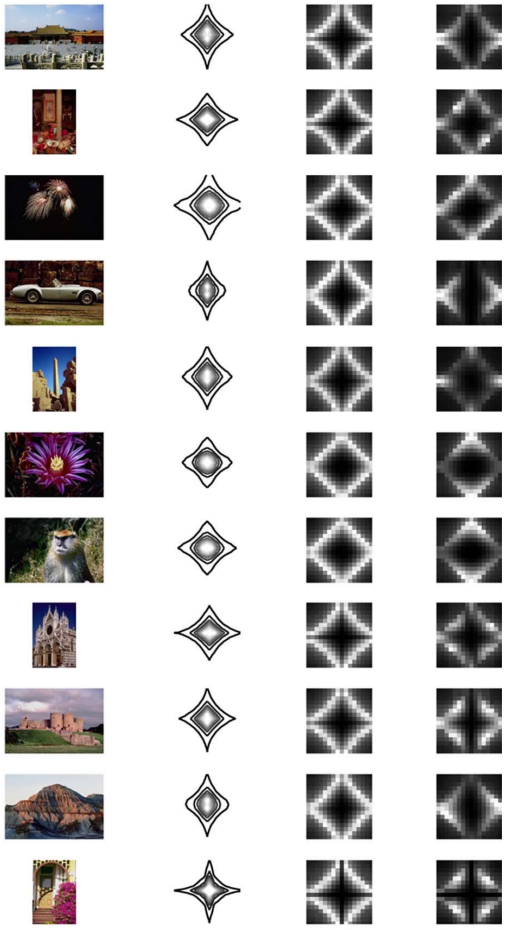


FIGURE II.9.: L'ACI produit des filtres plus adaptés aux différentes catégories d'images naturelles : (a) catégories d'images naturelles et (b) leur spectre. La colonne (c) montre les filtres obtenus par ACP : les filtres se ressemblent tous. La colonne (d) montre les filtres obtenus par ACI qui sont beaucoup plus spécifiques. Tiré de [LBGO07].

2.4. Conclusion

En résumé, pour la classification de scène, les approches holistiques basées sur les caractéristiques bas-niveau, particulièrement avec géométrie grossière et les approches statistiques, sont bien adaptées car elles capturent des informations qui sont par nature dispersées à travers toute l'image. Cependant, lorsque le nombre de classes augmente, il devient difficile de séparer les classes.

3. Approches basées sur les mots visuels

La deuxième grande famille d'approches utilise des techniques issues de l'analyse de texte pour classer les images. Plus précisément, une image est décrite au moyen de mots visuels, qui sont obtenus à partir de la classification non-supervisée de descripteurs locaux bas-niveau [4]. Ces méthodes utilisent donc une classification intermédiaire non-sémantique des caractéristiques locales. L'ensemble des mots est appelé le vocabulaire visuel.

Une fois le vocabulaire visuel obtenu, les images sont décrites à l'aide des mots visuels de la même manière qu'un texte est décrit par les mots de la langue naturelle. Dès lors, différentes techniques inspirées essentiellement de la recherche d'information textuelle sont appliquées.

Il existe cependant plusieurs limites à cette analogie (voir [Tir10, chap. 3] pour un développement complet). Une des plus importantes est qu'un texte est une chaîne linéaire de mots alors qu'une image est bi-dimensionnelle. Ainsi les mots entretiennent des relations géométriques. Il existe alors deux grandes familles d'approches :

1. Certaines approches modélisent explicitement les relations géométriques entre les mots (considérés comme des parties d'un objet). Ces méthodes sont de plus en plus populaires en reconnaissance d'objets mais n'ont pas été appliquées directement en reconnaissance de scènes. Une des raisons est que, du fait des déformations, les différentes instances d'une scène peuvent avoir des géométries différentes. De même, les différentes instances d'une catégorie de scènes peuvent aussi avoir des géométries très différentes. Nous en parlerons à la section II.4.2.2 dans le cadre des méthodes basées sur l'utilisation de la modélisation sémantique des scènes.
2. Les approches sans contraintes géométriques : dans ces approches, les relations géométriques sont ignorées : les images sont décrites par un ensemble non-ordonné de mots. Cette représentation est appelée « sac-de-mots » ou *Bag-of-Words (BoW)* en anglais. Le reste de cette section est dédié à la présentation des méthodes utilisant cette représentation.

Les méthodes BoW ont été développées en 2003 par Sivic et Zisserman [SZ03] dans un contexte de recherche d'images similaires (système *Video Google*), bien que certaines

4. Les mots visuels sont parfois appelés « textons ».

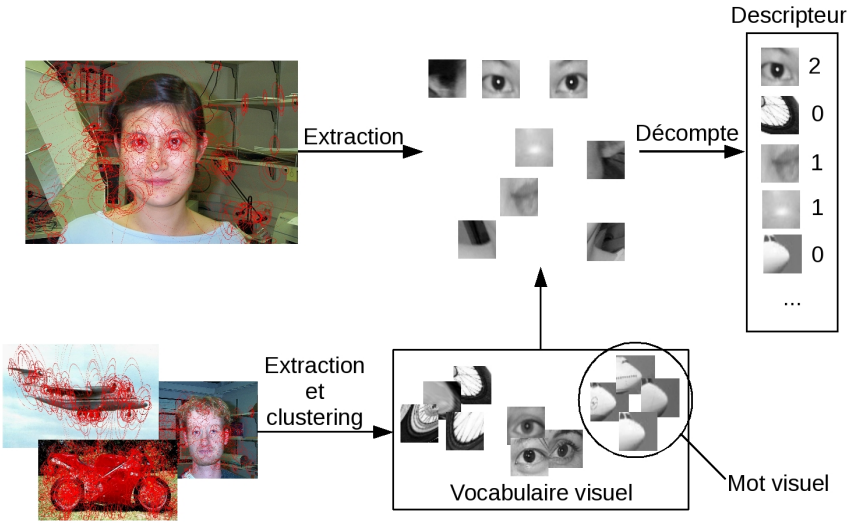


FIGURE II.10.: Illustration de l'approche sac de mots.

approches antérieures aient utilisé des techniques similaires. Le nom est choisi en référence directe aux techniques de recherche d'information dans les textes. Ces méthodes ont été utilisées avec succès dans des tâches variées en vision par ordinateur (classification d'objets, classification de scènes) et en recherche d'information (recherche d'images similaires, recherche d'images catégorisées).

3.1. Principe de base

La Figure II.10 présente le principe de base. Une fois le vocabulaire obtenu, une image est décrite de la manière suivante :

1. Les régions d'intérêt sont détectées dans l'image, puis décrites à l'aide de descripteurs locaux avec les algorithmes utilisés lors de la construction du vocabulaire
2. Chacun de ces descripteurs est associé au mot visuel du vocabulaire dont il est le plus proche : chaque région d'intérêt de l'image correspond ainsi à une occurrence d'un mot visuel
3. L'image est représentée par un vecteur, appelé « vecteur de fréquence », obtenu à partir du nombre d'occurrences de chaque mot dans l'image ; cela permet de se placer dans le cadre classique du modèle vectoriel en recherche d'information et en apprentissage (voir [SM83])

Le nom de la méthode vient du fait que la représentation finale obtenue ne tient pas compte de la position des mots dans l'image : les mots sont considérés de manière non-ordonnée, comme dans un sac.

Nous détaillons dans la suite les différents choix d'implémentation possibles pour les principaux composants d'un système BoW. Nous présentons tout d'abord les différents choix possibles de détecteurs de régions d'intérêt, de descripteurs et d'algorithmes de quantification vectorielle. Nous présentons ensuite les techniques de pondération qui permettent de construire le vecteur de fréquence. Enfin, nous présenterons les différentes techniques d'apprentissage utilisées.

3.1.1. Détecteurs de régions d'intérêt

La technique la plus utilisée est sans doute la détection de points invariants par affinité (voir Section I.2.3.3). Certains travaux utilisent plusieurs détecteurs de points d'intérêt : par exemple [SZ03] utilisent Harris-Affine et Maximally Stable Extremal Region (MSER). Ces détecteurs se focalisant sur des régions différentes, on accroît ainsi la robustesse.

Certaines études se basent aussi sur une segmentation en régions de l'image ([GA07]), voire sur l'étiquetage de chaque pixel ([WCM05]). Enfin, d'autres études ([FP05 ; AT06]) utilisent une grille régulière sur l'ensemble de l'image.

Les travaux de Nowak, Jurie et Triggs (voir [JT05 ; NJT06]) ont montré que, toutes choses étant égales par ailleurs, l'utilisation d'un échantillonnage aléatoire dense, à plusieurs échelles, est plus efficace dans une tâche de reconnaissance et de catégorisation d'objets. D'après les auteurs, le facteur-clé est le nombre de points détectés par image. Avec les méthodes basées sur les points d'intérêt, ce nombre est en général limité (de l'ordre de 10^3). Au contraire, les méthodes de sélection aléatoire permettent de sélectionner un nombre arbitrairement grand de points. Lorsqu'on se limite à quelques centaines de points par image, les méthodes basées sur les points d'intérêt sont, en général, plus performantes. Mais plus le nombre de points-clés par image augmente plus l'écart se resserre et finalement en utilisant un très grand nombre de points, les méthodes basées sur l'échantillonnage aléatoire sont plus performantes.

3.1.2. Descripteurs de régions d'intérêt

La descripteur Scale-Invariant Feature Transform (SIFT) est l'une des méthodes les plus utilisées (par exemple, [SZ03 ; Csu+04 ; NJT06]). Certains auteurs emploient des descripteurs beaucoup plus simples : « patches » d'image normalisés (par exemple [JT05]), méthodes basées sur des bancs filtres (par exemple [LM01 ; WCM05 ; VZ05]). Les travaux de Chen, Hu et Shen [CHS09] montrent cependant un avantage aux descripteurs locaux discriminants de type SIFT.

3.1.3. Algorithme de quantification vectorielle et construction du vocabulaire

L'algorithme le plus employé est sans doute les K-moyennes (par exemple, [SZ03 ; Csu+04]). Dans la plupart des études, la valeur de K (nombre de clusters) est choisie empiriquement,

pour maximiser les performances. Les valeurs obtenues pour K sont de l'ordre de quelques milliers (par exemple, dans [SZ03] $K = 6\ 000$ pour Harris-Affine, $K = 10\ 000$ pour MSER ; $K = 1\ 000$ dans [Csu+04]).

Les travaux de Nowak, Jurie et Triggs [NJT06] comparent l'utilisation des K-moyennes et de centres choisis au hasard parmi les points-clés obtenus. Les K-moyennes obtiennent un meilleur score mais la sélection de centres au hasard permet d'obtenir rapidement des résultats intéressants.

Choix du nombre de mots Le nombre de mots visuels est un paramètre important du système et son choix dépend de la tâche.

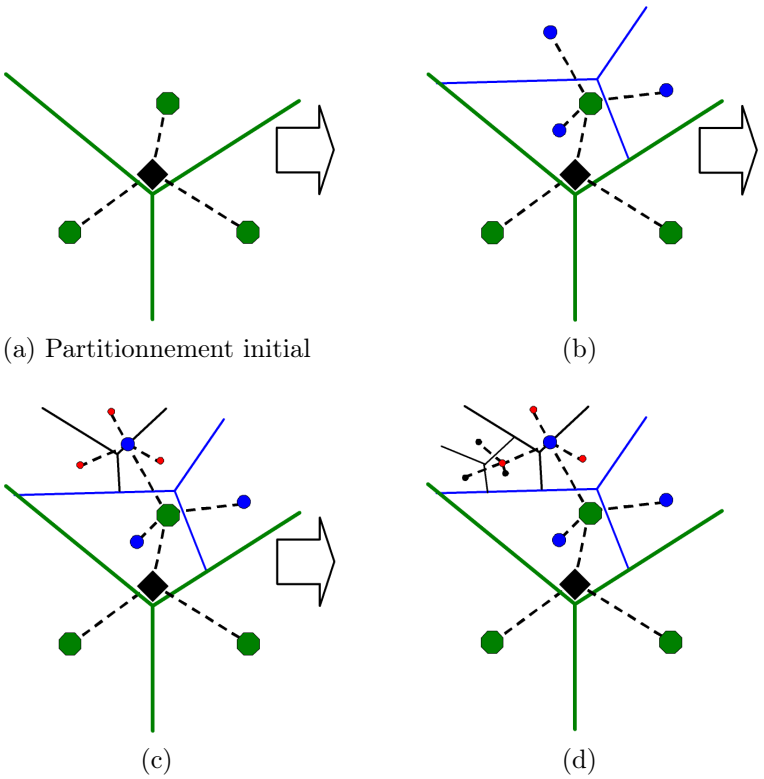
Le développement des méthodes BoW a entraîné l'apparition d'algorithmes de clustering rapides pour l'utilisation de grands dictionnaires. Par exemple Nistér et Stewénius [NS06] utilisent jusqu'à un million de mots visuels dans une tâche de recherche d'images. Ces algorithmes utilisent souvent une structure arborescente pour accélérer la recherche (voir Figure II.11). Dans l'approche de Nistér et Stewénius, l'algorithme des K-moyennes est tout d'abord appliqué sur l'ensemble des données, ce qui permet de les répartir en K groupes (voir II.11a). Le procédé est répété sur chacune des K régions jusqu'à obtenir un arbre de profondeur L. La recherche procède de manière récursive : on cherche d'abord le centre le plus proche au premier niveau, puis au niveau suivant, *etc.* La recherche consiste donc en $K \times L$ produits scalaires. Filliat [Fil07] développe une approche similaire en-ligne.

Winn, Criminisi et Minka [WCM05] part d'un dictionnaire large puis le réduit de manière supervisée, tout en maintenant la discriminabilité entre classes. Perronnin et al. [Per+06] partent d'un dictionnaire universel puis le spécialisent pour chaque classe. Une image est ensuite décrite par p histogrammes (un par classe), chacun étant classé indépendamment par une SVM.

Les méthodes qui sélectionnent automatiquement le nombre de clusters (voir Section I.3.1.2) sont peu utilisées car le coût calculatoire devient rédhibitoire quand le nombre de clusters dépasse quelques dizaines.

Fonction de distance Comme nous l'avons dit dans la Section I.3.1.2, le choix de la fonction de distance est un des critères fondamentaux lors de l'utilisation d'un algorithme de quantification vectorielle. Beaucoup d'auteurs utilisent, souvent implicitement, la distance ℓ_2 ou la distance de Mahalanobis. Nistér et Stewénius [NS06] comparent la distance ℓ_1 la distance ℓ_2 et trouvent que ℓ_1 donne de meilleurs résultats pour une tâche de recherche d'images. Tirilly, Claveau et Gros [TCG09] montrent que l'utilisation de distances fractionnaires (c'est-à-dire les distances ℓ_p où $p < 1$) augmente les performances, sans doute parce qu'elles donnent moins d'importance aux différences entre chaque coordonnée.

Beaucoup de descripteurs récents étant basés sur des histogrammes (SIFT, Histogram of



(a) Partitionnement initial (b)

(c) (d)

FIGURE II.11.: Principe de l'algorithme de clustering de [NS06].

Oriented Gradients (HOG)), l'utilisation de distances sur les histogrammes se développe. Wu [Wu09] montre qu'il est plus efficace de bâtir un dictionnaire en utilisant la distance d'intersection entre histogrammes que la distance ℓ_2 pour le descripteur CENTRIST.

Élimination des mots inutiles En recherche d'information textuelle, il peut s'avérer nécessaire d'ignorer certains mots (parfois appelés « mots vides »). Il existe deux approches principales pour sélectionner les mots à supprimer. Premièrement, des arguments linguistiques, permettent d'éliminer *a priori* certains mots très courants, comme les articles. Deuxièmement, l'hypothèse de Luhn (voir [Luh58]) stipule que les mots les plus fréquents et les moins fréquents peuvent être ignorés car soit ils apparaissent dans beaucoup de documents, soit ils ont peu de chances d'apparaître dans les requêtes. La liste des mots vides est appelée *stop-list*.

En se basant sur les mêmes arguments, Sivic et Zisserman [SZ03] éliminent 5% des mots les plus fréquents et 10% des mots les moins fréquents (ces seuils étant déterminés de manière empirique). Rappelons cependant que les mots visuels n'ont pas de sens donc les arguments linguistiques ne sont pas forcément justifiés.

Tirilly (voir [Tir10, chap. 3]) propose de constituer les *stops-lists* à l'aide de *Probabilistic Latent Semantic Analysis (pLSA)*, technique qui permet de regrouper les mots en thèmes (voir page 97) : les mots qui sont rares dans tous les thèmes sont éliminés. De plus, l'évaluation menée montre que la suppression des mots par la méthode de Luhn peut diminuer les résultats dans une tâche de recherche d'images.

3.1.4. Construction du vecteur de fréquence

La pondération des mots est une étape clé dans l'application des méthodes BoW en recherche d'information. En effet, l'hypothèse de base des méthodes BoW est que plus un terme est présent dans un document, mieux il le décrit. Cependant, deux effets sont à prendre en compte. Premièrement, si un terme apparaît dans un grand nombre de documents, il n'est pas spécifique à ces documents. Ce phénomène apparaît naturellement avec les articles, les pronoms, *etc.*, ce qui justifie de les éliminer *a priori* mais peut aussi apparaître avec des mots plus longs. Ainsi, dans un corpus de documents sur l'informatique, le terme « ordinateur » apparaîtra probablement dans un grand nombre de documents et ne sera pas très spécifique. Par opposition, le terme « cryptographie » n'apparaîtra que dans les documents relatifs à la sécurité informatique et sera donc très spécifique. Il peut donc s'avérer intéressant de favoriser les mots qui apparaissent avec une faible fréquence (cette hypothèse est complémentaire des techniques d'élimination de mots vues à la section précédente). Deuxièmement, plus un document est long, plus le nombre d'occurrences de chaque mot dans ce document est élevée. Prendre en compte la fréquence brute (nombre d'occurrences) des termes dans les documents aura donc pour effet de donner plus de poids aux documents longs. De manière générale, dans les systèmes BoW, un document ne peut pas être représenté uniquement par le nombre

d'occurrences de chaque mot : il est nécessaire de pondérer ce nombre pour obtenir le vecteur de fréquences.

Le schéma de pondération le plus utilisé est *Term Frequency - Inverse Document Frequency (tf-idf)* qui met directement en pratique les observations ci-dessus. Ainsi, afin de tenir compte de l'importance globale d'un terme t_i au niveau du corpus, on définit la fréquence documentaire inverse (*inverse document frequency* en anglais) idf_i de ce terme par :

$$idf_i = \log\left(\frac{\text{Nombre total de documents}}{\text{Nombre de documents où apparaît le terme } t_i}\right)$$

Les termes qui apparaissent souvent auront un *idf* petit (ils sont présents dans beaucoup de documents donc peu discriminants). À l'opposé, les termes rares auront un *idf* grand. Au final, l'importance du terme t_i au sein du document d_j est donnée par :

$$w_{ij} = t_{ij} \times idf_i$$

où t_{ij} est le nombre d'occurrences du terme t_i dans le document d_j (*term frequency* en anglais). Ainsi, les mots fréquents dans un document mais relativement rares dans le corpus (*idf* élevé) seront considérés comme très représentatifs.

Dans le domaine de la reconnaissance visuelle, Csurka et al. [Csu+04] utilisent directement le nombre d'occurrences. Sivic et Zisserman [SZ03] utilisent tf-idf. Une revue et une comparaison des performances des différents schémas de pondération se trouve dans Tirilly, Claveau et Gros [TCG09]. Ces résultats suggèrent qu'il n'y a pas de pondération optimale : celle-ci dépend du corpus, de la requête et de la mesure de distance utilisée. De plus, l'influence du schéma de pondération est faible sur les gros corpus.

3.1.5. Algorithme d'apprentissage

Une fois le vecteur de fréquences obtenu on a une représentation globale de l'image, indépendante du nombre de points d'intérêt trouvés. Ce vecteur de fréquence peut-être considéré comme un histogramme *i.e.* une approximation de la distribution des mots dans une image. On peut dès lors utiliser un classificateur pour apprendre une règle de décision dans une tâche de reconnaissance.

L'algorithme le plus employé est sans doute les SVM grâce à leurs excellentes performances. Csurka et al. [Csu+04] comparent l'utilisation des SVM et des Naive Bayes Classifiers (NBCs) : les SVM donnent de meilleurs résultats sur la tâche de reconnaissance de catégories. De plus, il existe des noyaux spécifiques pour les histogrammes. Une comparaison a été effectuée par Zhang et al. [Zha+07] : EMD est meilleur pour la catégorisation et χ_2 pour la reconnaissance de textures.

Cependant, il est possible d'utiliser des classificateurs beaucoup plus simples. Ainsi, dans l'article original, Sivic et Zisserman [SZ03] utilisent un classificateur basé sur les k-PPV avec la similarité du cosinus.

Globalement, les méthodes BoW obtiennent de très bons résultats dans de nombreuses tâches et sont simples à mettre en œuvre.

3.2. Avantages et inconvénients

D'un point de vue de la reconnaissance d'objets, la méthode BoW présente plusieurs avantages :

1. L'utilisation de descripteurs locaux très discriminants permet de prendre en compte des informations locales et fournit une représentation très précise de ces régions.
2. La méthode offre une représentation globale de l'image, de faible complexité pour les méthodes d'apprentissage et peu sensible au problème d'occlusion ; de plus, il est facile de combiner les informations en provenance de plusieurs vues de l'objet pour former une représentation invariante en point de vue.
3. La phase de quantification des descripteurs locaux permet d'obtenir plus de robustesse face au bruit et d'augmenter les facultés de généralisation.

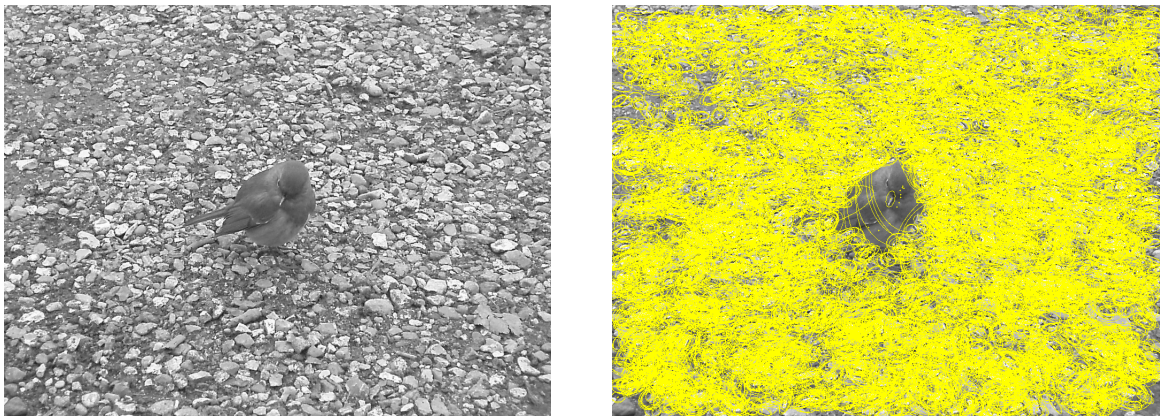
Du point de vue de la reconnaissance d'objets, le principal inconvénient de ces méthodes vient de l'absence d'information spatiale dans la représentation des images. Cependant, cette représentation désordonnée permet de s'affranchir des grandes variations géométriques entre les différentes instances d'une catégorie d'objets. Notons d'ailleurs qu'un des premiers articles sur les BoW traitait du problème de la catégorisation (voir [Csu+04]). Comme nous l'avons dit, la reconnaissance de scènes fait peu appel à des informations géométriques. Les méthodes BoW sont donc adaptées pour ce type de tâches.

Un autre inconvénient de ces méthodes est qu'il est difficile d'optimiser chaque étape : le choix des détecteurs influe sur le choix des descripteurs, qui influe sur l'algorithme de classification non-supervisée et la distance à utiliser. Ces choix influent enfin sur le schéma de pondération et de l'algorithme d'apprentissage à utiliser. La thèse de Pierre Tirilly ([Tir10]) et les travaux de Nowak, Jurie et Triggs [NJT06] et Yang et al. [Yan+07] fournissent des études détaillées de ces paramètres. Il en ressort qu'il n'y a pas de combinaison optimale : les choix dépendent fortement de la tâche à effectuer.

Une différence importante entre les méthodes BoW pour l'image et pour le texte se situe dans le fait que le vocabulaire d'un corpus de textes est issu directement du contenu des textes [5], alors qu'un vocabulaire visuel est construit de manière complètement artificielle. Le choix des détecteurs, des descripteurs et l'algorithme de classification non-supervisée influent directement sur la nature des mots visuels.

De plus, un mot est en général représentatif d'un concept alors qu'un mot visuel est caractéristique d'une région de l'image, indépendamment de son sens. Ainsi, le même mot peut représenter une partie d'un objet ou un élément du fond. Cela a deux conséquences :

5. À quelques prétraitements près, comme l'utilisation d'un racineur ou la détection de termes complexes remplaçant les termes de base.



(a) Image originale (b) Mots détectés

FIGURE II.12.: Exemple d'image où un grand nombre de mots d'arrière-plan sont détectés. Extrait de [Tir10, p. 66].

1. Les mots ne sont pas indépendants : comme un objet est décrit par plusieurs mots à chaque fois que l'objet est présent, l'ensemble de ces mots est présent aussi. Par exemple les mots correspondant aux différentes parties d'un visage sont toujours présents simultanément dans l'image.
2. La fréquence d'un mot n'est pas toujours liée à son importance sémantique. Par exemple, un même mot peut apparaître de nombreuses fois dans l'arrière fond d'une image (par exemple dans la figure II.12, de nombreux cailloux sont détectés). De plus, ce phénomène est sensible à l'échelle (dans une photo prise à une échelle moindre, les cailloux ne seraient peut-être pas détectés). Notons que ce problème se pose moins dans le cas des scènes. En effet, la détection de nombreux mots en arrière-plan comme dans la Figure II.12 indique clairement la composition de la scène.

Ainsi, l'hypothèse de base (une image peut être représentée par la proportion de mots visuels qui y apparaissent) ne va pas de soi pour les images. Cependant, les excellents résultats des méthodes BoW indiquent que l'hypothèse de base est valide.

3.3. Améliorations

Plusieurs améliorations ont été apportées à la technique de base que nous avons vue à la Section II.3.1.

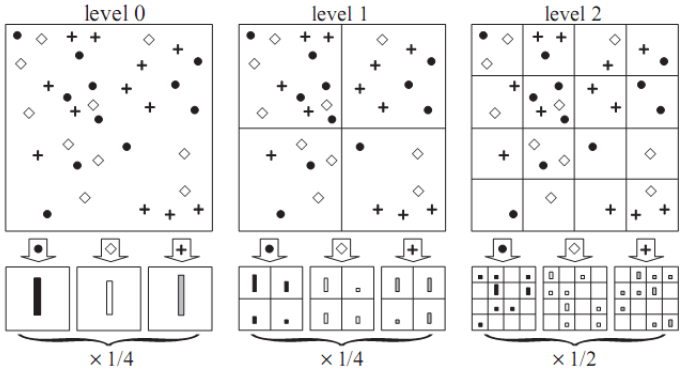
Incorporation de la géométrie Comme nous l'avons dit, il n'y a pas d'informations géométriques dans les représentations de type BoW. Plusieurs auteurs ont cherché à étendre

ces méthodes pour y ajouter des informations géométriques. Lazebnik, Schmidt et Ponce [LSP06] procèdent par découpages successifs de l'image : on construit d'abord un histogramme des mots visuels dans toute l'image puis l'image est divisée en quatre quadrants et on construit un histogramme pour chaque quadrant. Le processus est itéré sur chaque quadrant (voir figure II.13a). Le descripteur final (de très grande dimension) est obtenu en concaténant tous les histogrammes avec une pondération pour tenir compte de la taille du quadrant. Cette approche, baptisée *Spatial Pyramid Matching (SPM)*, permet de tenir compte de la géométrie. Notons cependant qu'elle est incompatible avec les techniques d'échantillonnage dense qui ont montré des résultats intéressants en catégorisation de scènes (en effet, on aurait toujours le même nombre de points d'intérêt par quadrant).

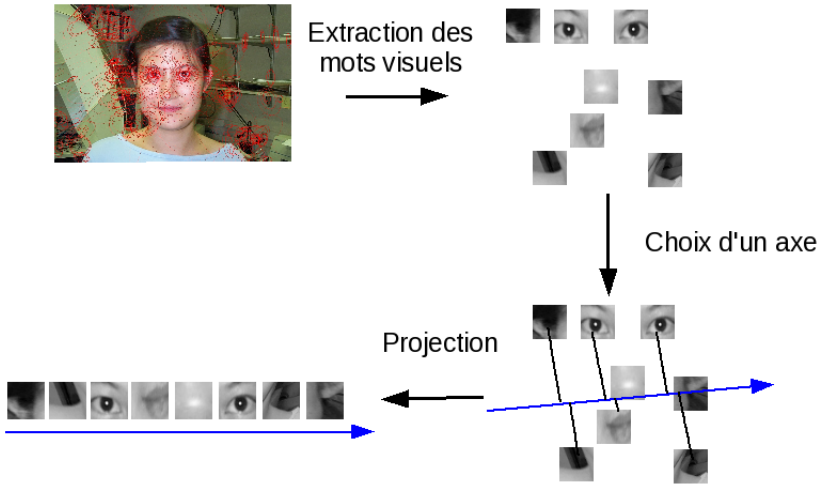
Cette technique a été utilisée par Quattoni et Torralba [QT09] spécialement pour le problème de classification des scènes d'intérieur, qui sont plus difficiles à catégoriser que les scènes d'extérieur. Pour cela, ils utilisent une base de données dans laquelle des régions d'intérêt (*i.e.* des objets) ont été définies manuellement pour former des images « prototypes » [6]. Les régions sont représentées par une approche de type SPM. De plus le descripteur GIST est utilisé pour comparer globalement les images. Les performances sur les images d'intérieur sont grandement améliorées par rapport au descripteur GIST seul. Cependant, cette approche nécessite d'avoir une base de données dans laquelle les objets ont été segmentés, ce qui est rarement disponible en pratique. De plus, cette méthode est confrontée aux problèmes de la variabilité des objets (voir Section II.1.2.2). Enfin, dans un environnement dynamique, les objets peuvent être déplacés,ce qui nuit sans doute à la méthode.

Une autre approche, proposée par Tirilly, Claveau et Gros [TCG08] permet d'utiliser la géométrie. Les points-clés sont extraits à l'aide du détecteur Hessian-Affine. L'idée principale consiste à utilisation de l'ACP sur la position de ces points afin de déterminer l'axe principal suivant lequel ces points sont distribués (voir figure II.13b). Les points-clés sont projetés sur cet axe, ce qui permet de définir naturellement un ordre entre les mots. Contrairement à la méthode classique qui ne prend en compte que la fréquence des mots, ici l'ordre des mots (visuels) est important. En poussant l'analogie avec le texte un cran plus loin, il est naturel d'utiliser les techniques de modélisation de la langue pour le problèmes de catégorisation. Cette technique est séduisante car elle permet de ramener la géométrie à des techniques bien connues de traitement de la langue. Nous verrons au chapitre III qu'elle ressemble à une des idées phares de notre travail. Cependant, l'extraction de l'axe principal n'est pas toujours simple en présence d'un arrière-plan complexe. De plus, cet axe dépend de l'angle de vue de l'objet.

6. L'identité des objets n'est pas utilisée donc ce ne sont pas des méthodes sémantiques. Seules des caractéristiques bas-niveau de ces régions et des mots visuels sont utilisés.



(a) [LSP06]



(b) [TCG08]

FIGURE II.13.: Illustration de deux approches incorporant des informations géométriques dans les techniques BoW.

Utilisation des modèles de thèmes Le deuxième type d'amélioration que nous présentons, là encore issu de la classification de documents textuels, est basé sur les modèles de thèmes (*topic model*). Ces techniques visent à découvrir des thèmes dans les documents afin d'améliorer leur classification : un thème est une distribution particulière de mots et un document est décrit comme un mélange probabiliste de thèmes (ce sont donc des modèles génératifs). Les modèles les plus employés sont *pLSA* (voir [Hof99]) et l'allocation de Dirichlet latente (*Latent Dirichlet Allocation (LDA)* voir [BNJ03]).

Par analogie, en reconnaissance visuelle, les thèmes sont des catégories d'objets (*e.g.* maison, ciel bleu, ciel nuageux) et les images sont décrites par ces thèmes (*e.g.* une image contient un ciel nuageux et de l'eau). Dans le cas de l'analyse de texte, les thèmes obtenus peuvent être interprétés *a posteriori* mais n'ont pas toujours un rapport avec le thème réel du document. En revanche, en reconnaissance visuelle, les thèmes n'ont pas d'interprétation (ce qui justifie que nous n'ayons pas classé ces méthodes parmi les méthodes sémantiques).

Quelhas et al. [Que+05] et Bosch, Zisserman et Muñoz [BZM06] ont appliqué pLSA au problème de la catégorisation de scènes. Une fois la représentation en thèmes obtenue, on entraîne un classificateur discriminatif pour distinguer les différentes classes. Fergus et al. [Fer+05] a étendu l'approche pLSA pour tenir compte des informations de position. Fei-Fei et Perona [FP05] utilisent LDA. Ici, les densités des thèmes sont propres à chaque classe. La classification se fait dans un cadre bayésien en choisissant la classe la plus probable d'une image de test. Les résultats obtenus sont très bons mais moins que les pyramides spatiales (voir [BMM07 ; Bos07]). Enfin, Bosch, Zisserman et Muñoz [BZM08] combinent les approches basées sur pLSA et les pyramides spatiales. Les résultats obtenus montrent une légère amélioration.

De manière générale, les extensions basées sur les modèles de thèmes ont apporté une faible amélioration au prix d'un temps de calcul beaucoup plus long. Les approches de type Spatial Pyramid Matching (SPM) permettent d'intégrer simplement une géométrie grossière dans le modèle BoW.

Nous présentons maintenant la dernière grande famille d'approches pour la reconnaissance de scènes, les approches sémantiques.

4. Approches basées sur la modélisation sémantique

L'idée générale de ces méthodes est d'effectuer une étape de classification intermédiaire qui modélise le contenu de l'image. Le résultat est ensuite utilisé pour la classification de la scène.

On distingue deux approches de modélisation intermédiaire :

1. on peut chercher à décrire l'image en terme de « composants » comme par exemple les objets qu'elle contient (voitures, personnes, *etc.*) ou la présence de régions spécifiques (ciel, herbe, *etc.*). Dans ce cas, les composants de l'image sont d'abord détectés et classifiés. Cette description est ensuite utilisée pour classer l'image dans son ensemble.

2. on peut aussi essayer de repérer des caractéristiques perceptuelles générales de l'image telles que l'ouverture (l'image contient-elle des structures éloignées ?), la rugosité, *etc.* L'image dans son ensemble est d'abord décrite en fonction de propriétés perceptuelles puis cette description est utilisée pour la classification.

Dans tous les cas, une fois la modélisation intermédiaire construite, on utilise un classificateur pour attribuer un label à l'image. Un certain nombre d'approches, notamment celles issues de la reconnaissance d'objets, utilisent des techniques de quantification de descripteurs locaux.

Nous présentons tout d'abord les méthodes basées sur la modélisation sémantique des régions (Section II.4.1) puis celles basées sur la modélisation des objets (Section II.4.2). Enfin, l'utilisation des propriétés perceptuelles est présentée à la Section II.4.3.

4.1. Modélisation sémantique des régions

Le principe de ces méthodes est de décrire une scène comme une collection de régions. Par exemple, une scène de plage sera décrite comme contenant des régions contenant de l'eau, des régions contenant du sable *etc.*Les travaux de [Rog+97 ; MGR04] ont identifié neuf catégories intermédiaires (appelées « concepts ») : « ciel », « eau », « herbe », « troncs », « feuillage », « champs », « pierres », « fleurs » et « sable ».

L'approche de Vogel et Schiele (voir [VS04 ; VS07]) est représentative de ces travaux. L'image est découpée en 10×10 régions (voir Figure II.14) et chaque région est décrite selon des critères bas-niveau (histogramme de couleur, histogramme d'orientation et critères de Haralick). Cette description bas-niveau est utilisée pour classer la régions dans une des catégories intermédiaires en utilisant des SVM ou un k-PPV. L'image est ensuite représentée par la proportion de chaque concept dans l'image (on notera la proximité avec les approches basées sur les mots visuels). Ce vecteur (appelé vecteur d'occurrence des concepts) est ensuite utilisé pour apprendre la classe de l'image (là aussi, on utilise des SVM).

D'autres approches utilisent une segmentation basée, en général, sur les informations de texture et de couleur pour détecter les régions ([MGR04]). Fan et al. [Fan+05] utilisent des fonctions de segmentation apprises à partir de l'ensemble d'apprentissage. Barnard et al. (voir [Bar+03]) utilisent une base de données d'images segmentées associée au texte environnant. Le texte est utilisé pour apprendre les catégories des régions et l'étiquette de l'image avec différents modèles de la densité jointe des mots et des images.

Aksoy et al. [Aks+05] ; Gokalp et Aksoy [GA07] et Singhal, Luo et Zhu [SLZ03] modélisent les relations spatiales entre les régions. Signalons que Shakey (voir section I.1) utilisait un tel système (voir Brice et Fennema [BF70]).

Enfin, il est possible d'utiliser à la fois des descripteurs bas-niveau et des descriptions sémantiques (voir par exemple le travail de Luo et Savakis [LS01] ; Luo, Savakis et

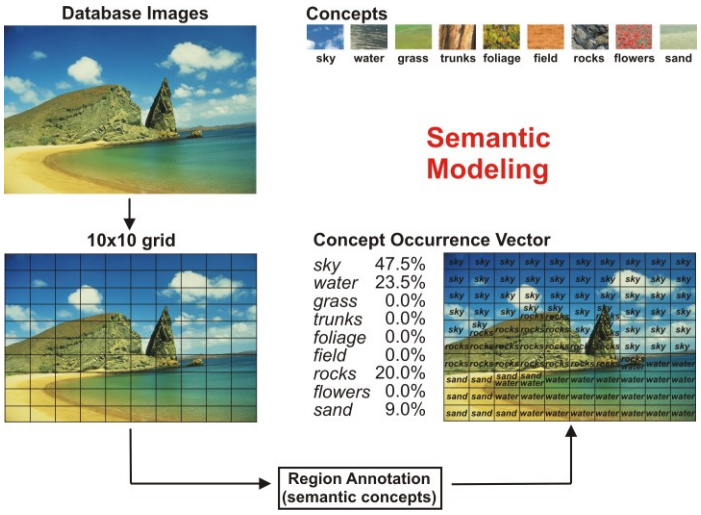


FIGURE II.14.: Approche utilisant la classification sémantique de régions. Extrait de [VS07].

SINGHAL [LSS05] ; VOGEL et al. [Vog+06] qui utilisent un réseau bayésien pour coder les relations entre ces deux niveaux de description).

4.2. Reconnaissance préalable des objets

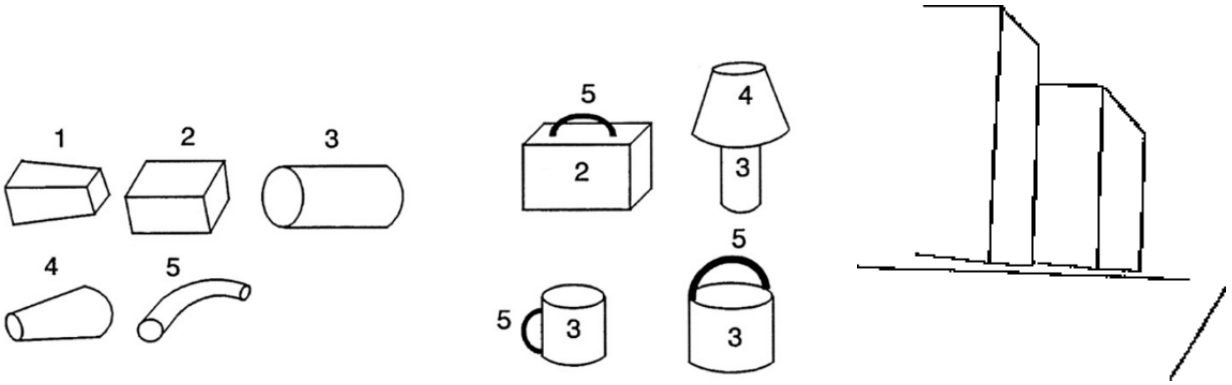
Historiquement, les premiers modèles d'analyse de scène se basaient sur la détection préalable des objets, comme par exemple les systèmes VISIONS (voir [HR78]) et CONDOR (voir [SF91]). De très nombreux travaux ont été proposés en reconnaissance d'objets et une présentation complète est hors du cadre de ce manuscrit.

La plupart des méthodes visent à reconnaître un exemplaire (une instance ou un exemplaire d'une catégorie) dans une image. Par opposition, décrire une scène en termes d'objets nécessite de localiser et reconnaître un nombre *a priori* inconnu d'objets.

Nous nous restreignons aux méthodes qui ont été appliquées dans le cadre de la reconnaissance de scènes, c'est-à-dire les méthodes de reconnaissance de catégories d'objets et renvoyons le lecteur à des ouvrages spécialisés ([Sze10, chap. 14]). On peut les séparer en plusieurs approches :

- Approches basées sur la forme : dans ces approches, on essaie de reconnaître la forme des objets directement à partir de l'image.
- Approches locales : ces approches sont basées sur l'extraction de caractéristiques locales qui vont servir à identifier les objets.

Notons ici qu'il n'y a pas d'approches globales (cela correspondrait à extraire une signature globale pour reconnaître des objets dans le but de classer la scène).



(a) Quelques géons.

(b) GSD de quelques objets (les numéros font référence au numéro des géons).

(c) Géons décrivant une scène de ville.

FIGURE II.15.: Illustration des approches basées sur la forme. Extrait de [Bie88].

4.2.1. Approches basées sur la forme

Approches structurelles Les approches structurelles (ou géométriques) prennent leur inspiration dans la théorie de la vision développées par David Marr (voir [Mar82]) qui postule que le but du système visuel humain est de reconstruire un modèle interne fin, en 3D, de la scène et des objets observés. Le modèle *Recognition by components (RBC)* de Biederman (voir [Bie87]) est largement inspiré de la troisième étape de Marr : Biederman définit un alphabet de 36 formes élémentaires qu'il nomme géons (voir II.15a). Les propriétés des géons (colinéarité, curvilinéarité, symétrie, parallélisme et co-terminaison) assurent qu'ils sont indépendants du point de vue et robustes aux transformations géométriques.

La reconnaissance a lieu en deux étapes : premièrement, les géons et les relations qui les lient sont extraits à partir de l'image (cette description est appelé *Geon Structural Description (GSD)*) ; deuxièmement la GSD de l'objet observé est comparée aux GSD des objets connus (voir II.15b).

BIEDERMAN [Bie88] a étendu sa théorie des géons à la reconnaissance de scènes. Pour tenir compte des relations entre les objets, Biederman introduit la notion de groupe de géons (*geon cluster*) qui est le plus gros volume comprenant deux ou trois géons disposés dans un arrangement familier (voir II.15c).

Approches géométriques Les techniques basées sur la forme et les contraintes géométriques (présentées au paragraphe II.2.2) ont été développées pour aborder les problèmes de la reconnaissance d'objets. Comme nous l'avons dit, ces approches nécessitent un modèle détaillé

de l'objet et ne sont donc pas adaptées à la catégorisation d'objets et donc à la description sémantique des scènes.

4.2.2. Approches locales

Principe Ces travaux sont en lien avec les approches basées sur les parties pour la reconnaissance d'objets (Fischler et Elschlager [FE73]). Dans ces approches, les différentes parties de l'objet sont modélisées et leurs relations sont encodées par un système de ressorts (voir figure II.16a). Le système est donc défini par un graphe. La reconnaissance consiste à minimiser l'énergie :

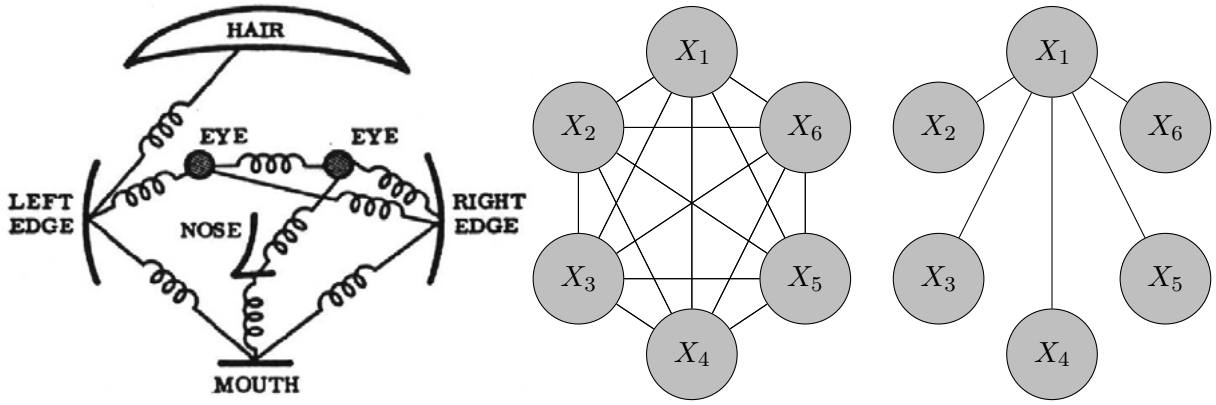
$$E = \sum_{i \in \mathcal{V}} V(\boldsymbol{x}_i) + \sum_{(i,j) \in \mathcal{E}} V(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{II.1}$$

pour toutes les positions potentielles des parties et des paires de parties reliées ($\mathcal{V}$ désigne l'ensemble des nœuds du graphe et $\mathcal{E}$ l'ensemble des arêtes). Cette énergie est similaire à celle utilisée dans les champs de Markov.

La complexité du problème général, appelé modèle en constellation (figure II.16b), est exponentielle : si N désigne le nombre de parties et P le nombre de positions possibles, trouver l'énergie minimale requiert de tester $\mathrm{O}\left(N^P\right)$ configurations. L'idée est alors de ne considérer que certaines dépendances possibles entre régions : on restreint donc les relations géométriques entre les parties. Différentes topologies sont possibles (voir [Fer05]), par exemple le modèle en étoile (figure II.16c) qui a une complexité en $\mathrm{O}\left(N^2 P\right)$.

Dans les approches modernes, les parties sont en fait des régions de l'espace détectées par des algorithmes de détection de régions d'intérêt (voir Section I.2.3.3). De plus de nombreux travaux utilisent une quantification des détecteurs bas-niveau. Ils peuvent donc aussi être vus comme une généralisation des approches BoW dans lesquelles on modélise les relations géométriques entre les mots. En se basant sur des caractéristiques locales et quantifiées, ces travaux obtiennent de très bons résultats dans les tâches de catégorisation d'objets (voir [Fer05]).

Application à la description de scènes Comme nous l'avons mentionné, en ce qui concerne la description de scènes, le problème est que le nombre d'objets et le nombre de parties d'objets sont inconnus. Les travaux de Sudderth et al. [Sud+05 ; Sud+08] sont représentatifs de cette approche. Le modèle est basé sur un ensemble de parties (analogues aux mots visuels) qui décrivent l'apparence locale et la position des objets (dans un repère centré sur l'objet). Les informations de localisation sont partagées entre les différentes images dans lesquelles un objet se trouve. Le modèle est hiérarchique : au niveau le plus haut, la distribution relative à la position des objets est modélisée ; au niveau plus bas, la distribution des parties dans un repère centré sur l'objet est modélisée (chaque catégorie d'objets est représentée par une



(a) Principe général (tiré de [FE73]) (b) Modèle en constellation (c) Modèle en étoile

FIGURE II.16.: Illustration des approches locales basées sur les parties d'objets. Chaque X_i représente une partie.

distribution sur l'ensemble des parties). Le processus de Dirichlet est utilisé pour modéliser la création des parties et des objets. L'apprentissage de ces distributions nécessite une grande base de données segmentée.

4.2.3. Conclusion sur les approches basées sur les objets

Globalement, les approches basées sur la détection d'objets sont intéressantes mais, en général, très complexes. Le problème de la catégorisation d'objets reste largement ouvert. De plus, de nombreuses études psychologiques ([Pot75 ; RJF05]) ont montré que la reconnaissance de scène peut se faire rapidement, sans passer par une étape de reconnaissance des objets. Ainsi, reposer sur cette étape pour catégoriser les scènes ne semble pas nécessaire. Les systèmes qui le font poursuivent en général un but au-delà de la classification.

4.3. Utilisation des propriétés perceptuelles

La dernière approche que nous développons se base sur la caractérisation sémantique de l'image dans son ensemble. Il n'y a donc pas de segmentation de l'image ou d'extraction des régions d'intérêt.

L'idée est de s'intéresser, dans un premier temps, aux critères perceptifs utilisés par les humains pour classer les images puis, dans un second temps, de construire une représentation informatique de ces critères. Ces travaux sont donc basés en partie sur des expériences de psychologie cognitive dans lesquelles on demande à des humains de regrouper spontanément des photos puis de décrire les critères employés.

Les travaux de Rogowitz et al. [Rog+97] ; Mojsilović, Gomes et Rogowitz [MGR04] suggèrent deux grands critères de classification : le degré de ressemblance à une forme humaine (qui sépare les formes humanoïdes des autres) et le degré de naturel (qui sépare les scènes de nature des scènes contenant de nombreux objets fabriqués par l'Homme). Cette approche a été étendue par Oliva et Torralba (voir [TO99 ; Oli+99 ; OT01]) qui proposent cinq axes de classification :

1. l'aspect naturel (*naturalness*), qui est lié aux orientations dominantes des contours : les environnements naturels apparaissent souvent texturés alors que les environnements humains sont dominés par des structures verticales et horizontales.
2. l'ouverture (*openness*), qui est liée au nombre de contours : les scènes ouvertes (côtes, autoroutes) sont caractérisées par leur faible nombre de contours (qui masquent éventuellement la ligne d'horizon) là ou les scènes contenant beaucoup de frontières sont dites fermées.
3. la rugosité (*roughness*), qui est liée à la taille des composants : une scène dont tous les contours sont petits est décrite comme rugueuse car elle contient beaucoup de petites structures.
4. l'expansion (*expansion*), qui est liée au nombre de lignes convergentes : les environnements humains contiennent de nombreuses lignes de fuite (intersection de contours) qui donnent la perception de la profondeur ; *a contrario*, une vue de face d'un immeuble aura une faible expansion.
5. l'aspérité (*ruggedness*), qui est liée aux contours obliques dans les images : les environnements humains ont en général peu d'aspérités là où une scène de montagne en a beaucoup.

La distinction la plus importante est l'aspect naturel ou non.

Pour estimer quantitativement ces grandeur perceptives, [TO99] propose d'utiliser le spectre. Leur technique, appelée patron spectral discriminant spectral, ou *Discriminant Spectral Template (DST)* en anglais. consiste à estimer la contribution de chaque fréquence à chacune des propriétés perceptuelles. L'estimation du DST pour chaque propriété se fait par une simple régression linéaire entre un ensemble d'images ordonnées par des humains selon le critère que l'on souhaite étudier et le spectre de ces images.

La réunion de ces propriétés constitue ce que les auteurs appellent l'enveloppe spatiale de la scène. La différence entre ces propriétés et des catégories classiques est que ces propriétés sont continues (sauf l'aspect naturel) : on peut donc organiser les scènes de manière continue le long de ces axes. Le descripteur GIST (voir page 36) est issu de ces travaux.

Les travaux de Guyader et al. [Guy+02] ainsi que la thèse de Hervé Le Borgne (voir [Le 04, chap. 4] ont développé ces travaux. En utilisant une technique de projection non-linéaire, ils extraient des clusters sémantiques qui leur permettent de définir une hiérarchie de concepts. Ces travaux montrent de plus que la couleur n'est pas nécessaire pour la catégorisation des images, bien qu'elle puisse aider.

Ces approches comportent des similarités avec les approches basées sur les attributs sémantiques des objets dont nous avons parlé plus haut.

5. Conclusion

La classification de scènes est un sujet relativement neuf en reconnaissance visuelle et de nombreuses approches ont été proposées.

Au niveau psychologique, les analyses holistiques (le *gist* d'une scène) et les analyses locales sont complémentaires. De plus, différents niveaux d'analyse sont possibles. Le premier niveau d'analyse intervient très rapidement dans le processus d'intégration des informations sensorielles. Il fournit une information sur la nature du contexte, en principe suffisante pour en identifier les propriétés fonctionnelles et catégoriques. Le deuxième niveau intervient plus tard et décompose la scène en objets. Cette description est utile pour l'interaction physique avec l'environnement (*e.g.* manipulation des objets) et permet des raisonnements spatiaux ou conceptuels de haut niveau.

Les approches globales basées sur les descripteurs bas-niveau sont rapidement limitées à cause du recouvrement des classes. L'introduction de géométrie grossière permet d'améliorer ces approches.

Les approches basées sur la modélisation sémantique essaient de diminuer le gouffre sémantique (voir section I.2) entre les catégories d'images et les descripteurs bas-niveau. Parmi celles-ci, les approches basées sur les objets sont, sans doute, trop complexes : l'identification des objets n'est donc pas un préalable pour la classification de scène. Les approches basées sur les régions et les propriétés sémantiques sont intéressantes mais elles reposent souvent sur des expériences psychophysiques, ce qui peut les rendre difficiles à exploiter complètement.

Les approches basées sur la quantification des descripteurs bas-niveau offrent un niveau de modélisation intermédiaire non-sémantique. Ces approches sont parmi les plus utilisées en reconnaissance visuelle car elles offrent de bonnes performances tout en étant relativement simples à mettre en œuvre. Notre travail sera principalement basé dessus.

Bibliographie

Reconnaissance visuelle

[Bie72] I. Biederman. « Perceiving real-world scenes ». Dans : *Science* 177.4043 (1972), p. 77–80 (cf. p. 81).

[Bie95] I. Biederman. « Visual object recognition ». Dans : *An invitation to cognitive science*. Sous la dir. de S. Kosslyn et D. Osherson. 2d Edition. MIT Press, 1995, p. 121–165 (cf. p. 73).

[BMM07] A. Bosch, X. Muñoz et R. Martí. « Which is the best way to organize/classify images by content ? » Dans : *Image and vision computing* 25 (2007), p. 778–791 (cf. p. 82, 99).

[BMR82] I. Biederman, R. Mezzanotte et J. Rabinowitz. « Scene perception : Detecting and judging objects undergoing relational violations ». Dans : *Cognitive Psychology* 14 (1982), p. 143–177 (cf. p. 77).

[Bou+04] M. Boutell et al. « Learning multi-label scene classification ». Dans : *Pattern Recognition* 37.9 (2004), 1757–1771 (cf. p. 80).

[Cot07] M. Cottret. « Exploration Visuelle d'Environnement Intérieur par Détection et Modélisation d'Objets Saillants ». Thèse de doct. Toulouse : Insitut National Polytechnique de Toulouse, 2007 (cf. p. 73).

[EK98] R. Epstein et N. Kanwisher. « A cortical representation of the local visual environment ». Dans : *Nature* 392.6676 (1998), p. 598–601 (cf. p. 77).

[Eps05] R. Epstein. « The cortical basis of visual scene processing ». Dans : *Real World Scene Perception*. Sous la dir. de J. M. Henderson. Taylor & Francis, 2005. Chap. 6, p. 954–978 (cf. p. 77).

[Far+09] A. Farhadi et al. « Describing Objects by their Attributes ». Dans : *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. T. 2. IEEE. Miami : IEEE Computer Society, juin 2009 (cf. p. 76).

[Fei+05] L. Fei-Fei et al. « Why does natural scene categorization requires little attention ? Exploring attentional requirements and neurophysiological evidence ». Dans : *Real World Scene Perception*. Sous la dir. de J. M. Henderson. Taylor & Francis, 2005. Chap. 3, p. 893–924 (cf. p. 77).

[Fer05] R. Fergus. « Visual Object Category Recognition ». Thèse de doct. Oxford : University of Oxford, 2005, p. 193 (cf. p. 74–76, 103).

[FFT07] L. Fei-Fei, R. Fergus et A. Torralba. *Recognizing and learning object categories*. 2007 (cf. p. 75).

[GHP07] G. Griffin, A. Holub et P. Perona. *Caltech-256 Object Category Dataset*. Rap. tech. 7694. California Institute of Technology, 2007 (cf. p. 71).

[Hen05b] J. Henderson. « Introduction to real-world scene perception ». Dans : *Real World Scene Perception*. Sous la dir. de J. M. Henderson. Taylor & Francis, 2005, p. 849–851 (cf. p. 70).

[Mar82] D. Marr. *Vision : A computational investigation into the human representations and processing of visual information*. San Francisco : Freeman, 1982 (cf. p. 81, 102).

[Oli05] A. Oliva. « Gist of the scene ». Dans : *Neurobiology of attention*. Sous la dir. de L. Itti, G. Rees et J. Tsotsos. Academic Press, 2005. Chap. 41, p. 251–257 (cf. p. 81).

[OS97] A. Oliva et P. Schyns. « Coarse blobs or fine edges ? Evidence that information diagnosticity changes the perception of complex visual stimuli ». Dans : *Cognitive Psychology* 34 (1997), p. 72–107 (cf. p. 81).

[OT01] A. Oliva et A. Torralba. « Modeling the shape of the scene : A holistic representation of the spatial envelope ». Dans : *International Journal of Computer Vision* 42.3 (2001), p. 145–175 (cf. p. 70, 71, 105).

[OT06] A. Oliva et A. Torralba. « Building the gist of a scene : The role of global image features in recognition ». Dans : *Progress in brain research* 155 (2006), p. 23–36 (cf. p. 81).

[OT07] A. Oliva et A. Torralba. « The role of context in object recognition ». Dans : *Trends in Cognitive Sciences* 11.12 (2007), p. 520–527 (cf. p. 78).

[Pot75] M. Potter. « Meaning in visual search ». Dans : *Science* 187 (1975), 965–966 (cf. p. 77, 104).

[RJF05] G. Rousselet, O. Joubert et M. Fabre-Thorpe. « How long to get to the "gist" of real-world natural scenes ? » Dans : *Real World Scene Perception*. Sous la dir. de J. M. Henderson. Taylor & Francis, 2005. Chap. 1, p. 852–877 (cf. p. 77, 104).

[Sze10] R. Szeliski. *Computer Vision, Algorithms and Applications*. Texts in Computer Science. Springer, 2010 (cf. p. 69, 101).

[TH83] B. Tversky et K. Hemenway. « Categories of environmental scenes ». Dans : *Cognitive psychology* 15.1 (1983), p. 121–149 (cf. p. 77).

[TO03] A. Torralba et A. Oliva. « Statistics of natural image categories ». Dans : *Network : Computation in Neural Systems* 14 (2003), p. 391–412 (cf. p. 77, 79).

[Tor03] A. Torralba. « Contextual priming for object detection ». Dans : *International Journal of Computer Vision* 53.2 (2003), p. 169–191 (cf. p. 78).

[Vog+06] J. Vogel et al. « Categorization of natural scenes : local vs. global information ». Dans : *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*. 2006 (cf. p. 81, 83, 101).

Approches basées sur les caractéristiques bas-niveau

[BBV02] S. Boughorbel, N. Boujemaa et C. Vertan. « Histogram-based color signatures for image indexing ». Dans : *Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Citeseer. 2002 (cf. p. 82).

[Dou+09] M. Douze et al. « Evaluation of GIST descriptors for web-scale image search ». Dans : *Proceedings of the 8th ACM International Conference on Image and Video Retrieval (CIVR 2009)*. Sous la dir. de S. Marchand-Maillet et Y. Kompatsiaris. Santorini Island, Greece : ACM, juil. 2009. isbn : 978-1-60558-480-5 (cf. p. 85).

[GP94] M. Gorkani et R. Picard. « Texture orientation for sorting photos "at a glance" ». Dans : *IEEE Conference on Pattern Recognition*. T. 1. Jerusalem, 1994, 459–464 (cf. p. 84).

[Hat92] J. H. van Hateren. « Theoretical predictions of spatiotemporal receptive fields of fly LMCs, and experimental validation ». Dans : *Journal of Comparative Physiology* A 171 (1992), p. 157–170 (cf. p. 86).

[HCO99] A. Hyvärinen, R. Cristescu et E. Oja. « A fast algorithm for estimating overcomplete ICA bases for image windows ». Dans : *International Joint Conference on Neural Networks*. 1999 (cf. p. 86).

[HR04] P. Howarth et S. Rüger. « Evaluation of Texture Features for Content-Based Image Retrieval ». Dans : *Lecture Notes in Computer Science* 3115 (2004), p. 2134–2135 (cf. p. 82).

[HS98] J. van Hateren et A. van der Schaaf. « Independent component filters of natural images compared with simple cells in primary visual cortex ». Dans : *Proc R Soc Lond [Biol]* 265.1394 (1998). Proceedings of the Royal Society of London Series B - Biological Sciences, p. 359–366 (cf. p. 86).

[LBGO07] H. Le Borgne, A. Guérin-Dugué et N. O'Connor. « Learning midlevel image features for natural scene and texture classification ». Dans : *Circuits and Systems for Video Technology, IEEE Transactions on* 17.3 (2007), p. 286–297 (cf. p. 86, 87).

[LBP99] A. Labbi, H. Bosch et C. Pellegrini. « Image categorization using independent component analysis ». Dans : *ACAI Workshop on Biologically Inspired Machine Learning (BIML 1999)*. Citeseer. 1999 (cf. p. 86).

[Le 04] H. Le Borgne. « Analyse de scènes naturelles par composantes indépendantes ». Thèse de doct. Grenoble : Institut national polytechnique, 2004, p. 190 (cf. p. 86, 105).

[LJJ08] J. Lee, A. Jain et R. Jin. « Scars, marks and tattoos (SMT) : Soft biometric for suspect and victim identification ». Dans : *Proceedings of the 2008 Biometrics Symposium*. 2008, p. 1–8 (cf. p. 85).

[LL03] S. Lim et G. Lu. « Effectiveness and Efficiency of Six Colour Spaces for Content Based Image Retrieval ». Dans : *Third International Workshop on Content-Based Multimedia Indexing*. 2003 (cf. p. 82).

[Low04] D. G. Lowe. « Distinctive Image Features from Scale-Invariant Keypoints ». Dans : *International Journal of Computer Vision* 60.2 (2004), p. 91–110 (cf. p. 85).

[Low99] D. G. Lowe. « Object recognition from local scale-invariant features. » Dans : *International Conference on Computer Vision*. T. 2. 1999, p. 1150–1157 (cf. p. 85).

[Man+01] B. Manjunath et al. « Color and texture descriptors ». Dans : *Circuits and Systems for Video Technology, IEEE Transactions on* 11.6 (2001), p. 703–715 (cf. p. 85).

[MZ98] W. Ma et H. Zhang. « Benchmarking of image features for content-based retrieval ». Dans : *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*. T. 1. Ieee. 1998, p. 253–257 (cf. p. 82).

[SB91] M. J. Swain et D. H. Ballard. « Color indexing ». Dans : *International Journal of Computer Vision* 7 (1991), 11–32 (cf. p. 82).

[SM97] C. Schmid et R. Mohr. « Local grayvalue invariants for image retrieval ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.5 (1997), p. 530–535 (cf. p. 85).

[SP98] M. Szummer et R. Picard. « Indoor-outdoor image classification ». Dans : *Proceedings of the 1998 IEEE International Workshop on Content-Based Access of Image and Video Database (CAIVD 1998)*. IEEE. Bombay, India, jan. 1998, p. 42–51 (cf. p. 84).

[SSL02] N. Serrano, A. Savakis et J. Luo. « A computationally efficient approach to indoor/outdoor scene classification ». Dans : *International Conference on Pattern Recognition*. Quebec, 2002 (cf. p. 84).

[SSN05] J. Shen, J. Shepherd et A. H. H. Ngu. « Semantic-Sensitive Classification for Large Image Libraries ». Dans : *Proceedings of the 11th International Multimedia Modelling Conference (MMM 2005)*. 2005 (cf. p. 82).

[TP91] M. Turk et A. Pentland. « Eigenfaces for recognition ». Dans : *Journal of Cognitive Neuroscience* 3.1 (1991), p. 71–86 (cf. p. 86).

[Vai+01] A. Vailaya et al. « Image Classification for Content-Based Indexing ». Dans : *IEEE Transactions on Image Processing* 10.1 (2001), p. 117–130 (cf. p. 82).

[VJZ98] A. Vailaya, A. Jain et H. J. Zhang. « On image classification : city vs. landscape ». Dans : *Pattern Recognition* 31 (1998), 1921–1935 (cf. p. 82, 84).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 85, 93).

Approches basées sur les mots visuels

[AT06] A. Agarwal et B. Triggs. « Hyperfeatures – multilevel local coding for visual recognition ». Dans : *European Conference on Computer Vision* (2006) (cf. p. 90).

[BMM07] A. Bosch, X. Muñoz et R. Martí. « Which is the best way to organize/classify images by content ? » Dans : *Image and vision computing* 25 (2007), p. 778–791 (cf. p. 82, 99).

[BNJ03] D. Blei, A. Ng et M. Jordan. « Latent Dirichlet Allocation ». Dans : *Journal of Machine Learning Research* 3 (2003), p. 993–1022 (cf. p. 99).

[Bos07] A. Bosch. « Image Classification for a large number of object categories ». Thèse de doct. Universitat de Girona : Departament d'Electrònica, Informàtica i Automàtica, sept. 2007 (cf. p. 99).

[BZM06] A. Bosch, A. Zisserman et X. Muñoz. « Scene Classification via PLSA ». Dans : *European Conference on Computer Vision*. T. 4. Graz, Austria, 2006, p. 517–530 (cf. p. 99).

[BZM08] A. Bosch, A. Zisserman et X. Muñoz. « Scene classification using a hybrid generative/discriminative approach ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008), p. 712–727 (cf. p. 99).

[CHS09] X. Chen, X. Hu et X. Shen. « Spatial weighting for bag-of-visual-words and its application in content-based image retrieval ». Dans : *Advances in Knowledge Discovery and Data Mining* (2009), p. 867–874 (cf. p. 90).

[Csu+04] G. Csurka et al. « Visual categorization with bags of keypoints ». Dans : *Workshop on Statistical Learning in Computer Vision, ECCV*. 2004, p. 1–22 (cf. p. 90, 91, 94, 95).

[Fer+05] R. Fergus et al. « Learning Object Categories from Google's Image Search ». Dans : *Procceding of the Tenth International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*. T. 2. IEEE Computer Society, oct. 2005, p. 1816–1823. isbn : 0-7695-2334-X (cf. p. 99).

[Fil07] D. Filliat. « A visual bag of words method for interactive qualitative localization and mapping ». Dans : *Proceedings of the International Conference on Robotics and Automation (ICRA)*. 2007 (cf. p. 91).

[FP05] L. Fei-Fei et P. Perona. « A Bayesian Hierarchical Model for Learning Natural Scene Categories ». Dans : *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. T. 2. 2005, p. 524–531 (cf. p. 90, 99).

[GA07] D. Gokalp et S. Aksoy. « Scene Classification Using Bag-of-Regions Representations ». Dans : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*. Minneapolis, USA, 2007, p. 1–8 (cf. p. 90, 100).

[Hof99] T. Hofmann. « Probabilistic latent semantic indexing ». Dans : *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 1999, p. 50–57 (cf. p. 99).

[JT05] F. Jurie et B. Triggs. « Creating efficient codebooks for visual recognition ». Dans : *Procceding of the Tenth International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*. T. 1. IEEE Computer Society, oct. 2005. isbn : 0-7695-2334-X (cf. p. 90).

[LM01] T. Leung et J. Malik. « Representing and recognizing the visual appearance of materials using three-dimensional textons ». Dans : *International Journal of Computer Vision* 43.1 (2001), p. 29–44 (cf. p. 90).

[LSP06] S. Lazebnik, C. Schmidt et J. Ponce. « Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories ». Dans : *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*. T. 2. IEEE. New-York : IEEE Computer Society, juin 2006, p. 2169–2178 (cf. p. 97, 98).

[Luh58] H. Luhn. « The automatic creation of literature abstracts ». Dans : *IBM Journal of research and development* 2.2 (1958), p. 159–165 (cf. p. 93).

[NJT06] E. Nowak, F. Jurie et B. Triggs. « Sampling strategies for bag-of-features image classification ». Dans : *Lecture Notes in Computer Science* 3954 (2006), p. 490 (cf. p. 90, 91, 95).

[NS06] D. Nistér et H. Stewénius. « Scalable recognition with a vocabulary tree ». Dans : *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*. T. 2. IEEE. New-York : IEEE Computer Society, juin 2006, p. 2161–2168 (cf. p. 91, 92).

[Per+06] F. Perronnin et al. « Adapted vocabularies for Generic Visual Categorization ». Dans : *Proceedings of ECCV 2006*. 2006 (cf. p. 91).

[QT09] A. Quattoni et A. Torralba. « Recognizing Indoor Scenes ». Dans : *IEEE Conference on Computer Vision and Pattern Recognition*. 2009 (cf. p. 97).

[Que+05] P. Quelhas et al. « Modeling scenes with local descriptors and latent aspects ». Dans : *Procceding of the Tenth International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*. T. 1. IEEE. IEEE Computer Society, oct. 2005, p. 883–890. isbn : 0-7695-2334-X (cf. p. 99).

[SM83] G. Salton et M. McGill. *Modern Information Retrieval*. McGraw-Hill, 1983 (cf. p. 89).

[SZ03] J. Sivic et A. Zisserman. « Video Google : A Text Retrieval Approach to Object Matching in Videos ». Dans : *International Conference on Computer Vision*. T. 2. 2003, p. 1470–1477 (cf. p. 88, 90, 91, 93, 94).

[TCG08] P. Tirilly, V. Claveau et P. Gros. « Language modeling for bag-of-visual words image categorization ». Dans : *Proceedings of the 7th ACM International Conference on Image and Video Retrieval (CIVR 2008)*. Sous la dir. de J. Luo et al. Niagara Falls, Ontario, Canada : ACM, juil. 2008, p. 249–258. isbn : 978-1-60558-480-5 (cf. p. 97, 98).

[TCG09] P. Tirilly, V. Claveau et P. Gros. *A review of weighting schemes for bag of visual words image retrieval*. Rap. tech. IRISA, 2009 (cf. p. 91, 94).

[Tir10] P. Tirilly. « Traitement automatique des langues pour l'indexation d'images ». Thèse de doct. Rennes, France : Université de Rennes 1, juil. 2010 (cf. p. 88, 93, 95, 96).

[VZ05] M. Varma et A. Zisserman. « A statistical approach to texture classification from single images ». Dans : *International Journal of Computer Vision : Special Issue on Texture Analysis and Synthesis* 62.1-2 (2005), 61–81 (cf. p. 90).

[WCM05] J. Winn, A. Criminisi et T. Minka. « Object categorization by learned universal visual dictionary ». Dans : *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*. T. 2. IEEE. 2005, p. 1800–1807 (cf. p. 90, 91).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 85, 93).

[Yan+07] J. Yang et al. « Evaluating bag-of-visual-words representations in scene classification ». Dans : *Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR 2007)*. Sous la dir. de J. Z. Wang et al. ACM. Augsburg, Bavaria, Germany, sept. 2007, p. 197–206 (cf. p. 95).

[Zha+07] J. Zhang et al. « Local features and kernels for classification of texture and object categories : a comprehensive study ». Dans : *International Journal of Computer Vision* 73.2 (2007), 213–238 (cf. p. 94).

Approches basées sur la modélisation sémantique

[Aks+05] S. Aksoy et al. « Learning Bayesian classifiers for scene classification with a visual grammar ». Dans : *Geoscience and Remote Sensing, IEEE Transactions on* 43.3 (2005), p. 581–589 (cf. p. 100).

[Bar+03] K. Barnard et al. « Matching words and pictures ». Dans : *Journal of Machine Learning Research* 3 (2003), p. 1107–1135 (cf. p. 100).

[BF70] C. Brice et C. Fennema. « Scene analysis using regions ». Dans : *Artificial intelligence* 1.3–4 (1970), p. 205–226 (cf. p. 100).

[Bie87] I. Biederman. « Recognition by components : A theory of human image understanding ». Dans : *Psychological Review* 94 (1987), p. 115–147 (cf. p. 102).

[Bie88] I. Biederman. « Aspects and extension of a theory of human image understanding ». Dans : *Computational Processes in Human Vision : An Interdisciplinary Perspective*. Norwood, New Jersey : Ablex Publishing Corporation, 1988 (cf. p. 102).

[Fan+05] J. Fan et al. « Statistical modeling and conceptualization of natural images ». Dans : *Pattern Recognition* 38.6 (2005), p. 865–885 (cf. p. 100).

[FE73] M. Fischler et R. Elschlager. « The representation and matching of pictorial structures ». Dans : *IEEE Transactions on Computer* C-22.1 (1973), 67–92 (cf. p. 103, 104).

[Fer05] R. Fergus. « Visual Object Category Recognition ». Thèse de doct. Oxford : University of Oxford, 2005, p. 193 (cf. p. 74–76, 103).

[GA07] D. Gokalp et S. Aksoy. « Scene Classification Using Bag-of-Regions Representations ». Dans : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*. Minneapolis, USA, 2007, p. 1–8 (cf. p. 90, 100).

[Guy+02] N. Guyader et al. « Towards the introduction of human perception in a natural scene classification system ». Dans : *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*. IEEE. 2002, p. 385–394 (cf. p. 105).

[HR78] A. Hanson et E. Riseman. « VISIONS : A computer system for interpreting scenes ». Dans : *Computer Vision Systems*. Sous la dir. d'A. Hanson et E. Riseman. New York : Academic Press, 1978, 303–333 (cf. p. 101).

[Le 04] H. Le Borgne. « Analyse de scènes naturelles par composantes indépendantes ». Thèse de doct. Grenoble : Institut national polytechnique, 2004, p. 190 (cf. p. 86, 105).

[LS01] J. Luo et A. Savakis. « Indoor vs outdoor classification of consumer photographs using low-level and semantic features ». Dans : *Image Processing, 2001. Proceedings. 2001 International Conference on*. T. 2. IEEE. 2001, p. 745–748 (cf. p. 100).

[LSS05] J. Luo, A. Savakis et A. Singhal. « A Bayesian network-based framework for semantic image understanding ». Dans : *Pattern Recognition* 38.6 (2005), p. 919–934 (cf. p. 100).

[Mar82] D. Marr. *Vision : A computational investigation into the human representations and processing of visual information*. San Francisco : Freeman, 1982 (cf. p. 81, 102).

[MGR04] A. Mojsilović, J. Gomes et B. Rogowitz. « Semantic-friendly indexing and quering of images based on the extraction of the objective semantic cues ». Dans : *International Journal of Computer Vision* 56.1 (2004), p. 79–107 (cf. p. 100, 105).

[Oli+99] A. Oliva et al. *Global Semantic Classification of Scenes using Power Spectrum Templates*. Challenges of Image Retrieval (CIR 1999). 1999 (cf. p. 105).

[OT01] A. Oliva et A. Torralba. « Modeling the shape of the scene : A holistic representation of the spatial envelope ». Dans : *International Journal of Computer Vision* 42.3 (2001), p. 145–175 (cf. p. 70, 71, 105).

[Pot75] M. Potter. « Meaning in visual search ». Dans : *Science* 187 (1975), 965–966 (cf. p. 77, 104).

[RJF05] G. Rousselet, O. Joubert et M. Fabre-Thorpe. « How long to get to the "gist" of real-world natural scenes ? » Dans : *Real World Scene Perception*. Sous la dir. de J. M. Henderson. Taylor & Francis, 2005. Chap. 1, p. 852–877 (cf. p. 77, 104).

[Rog+97] B. Rogowitz et al. « Perceptual image similarity experiments ». Dans : *Proceedings of the SPIE Conference on Human Vision and Electronic Imaging (HVIE 1997)*. 1997, p. 576–590 (cf. p. 100, 105).

[SF91] T. Strat et M. Fischler. « Context-based vision : Recognizing objects using information from both 2-D and 3-D imagery ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.10 (1991), 1050–1065 (cf. p. 101).

[SLZ03] A. Singhal, J. Luo et W. Zhu. « Probabilistic spatial context models for scene content understanding ». Dans : *Conference on Computer Vision and Pattern Recognition*. T. 1. 2003, 235–241 (cf. p. 100).

[Sud+05] E. B. Sudderth et al. « Learning Hierarchical Models of Scenes, Objects, and Parts ». Dans : *International Conference on Computer Vision*. T. 2. 2005, 331–1338 (cf. p. 103).

[Sud+08] E. B. Sudderth et al. « Describing visual scenes using transformed objects and parts ». Dans : *International Journal of Computer Vision* 77.1-3 (2008), p. 291–330 (cf. p. 103).

[Sze10] R. Szeliski. *Computer Vision, Algorithms and Applications*. Texts in Computer Science. Springer, 2010 (cf. p. 69, 101).

[TO99] A. Torralba et A. Oliva. « Semantic organization of scenes using discriminant structural templates ». Dans : *Proceedings of the Seventh International Conference On Computer Vision (ICCV 1999), 20-25 September, 1999, Kerkyra, Corfu, Greece*. T. 2. Published by the IEEE Computer Society. 1999, p. 1253–1260 (cf. p. 105).

[Vog+06] J. Vogel et al. « Categorization of natural scenes : local vs. global information ». Dans : *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*. 2006 (cf. p. 81, 83, 101).

[VS04] J. Vogel et B. Schiele. « Natural scene retrieval based on a semantic modeling step ». Dans : *Proceedings of International Conference on Image and Video Retrieval*. Dublin, 2004 (cf. p. 100).

[VS07] J. Vogel et B. Schiele. « Semantic Modeling of Natural Scenes for Content-Based Image Retrieval ». Dans : *International Journal of Computer Vision* 72.2 (2007), p. 133–157 (cf. p. 100, 101).

III. Reconnaissance de lieux sémantiques en robotique

Le chapitre précédent nous a permis d'aborder le problème de la reconnaissance visuelle par le truchement de la reconnaissance de scènes. Nous présentons dans ce chapitre le problème de la reconnaissance de lieux sémantiques par un robot mobile et les travaux dans ce domaine. La section III.1 est consacrée à la définition du concept de lieu sémantique et à l'utilité de la reconnaissance de lieux pour la robotique. La section III.2 est consacrée à la différence entre la reconnaissance de lieux sémantiques et les travaux en SLAM et en reconnaissance de scènes. La section III.3 décrit les difficultés spécifiques de la tâche de reconnaissance de lieux. L'état de l'art est présenté dans la section III.4.

1. Introduction

1.1. Le concept de lieu sémantique

Nous avons déjà rencontré le concept de lieu dans la partie sur la navigation (voir Section I.1.2) et plus particulièrement sur les cartes topologiques. Nous avons noté que les approches cartographiques réduisent le lieu à un repère dans l'espace, défini à des fins de navigation, sans information sémantique (c'est pourquoi nous préférons les nommer « nœuds de la carte topologique »).

À l'opposé, nous sommes intéressés par la reconnaissance de lieux tels « le bureau de Malika », « le couloir du 2^{e} étage du LIMSI », *etc.* Ces lieux sont des entités possédant une extension spatiale et définis par les êtres humains en fonction de leur unité fonctionnelle et sémantique. Ils correspondent aux divisions habituelles de l'espace à l'intérieur d'un bâtiment. C'est pourquoi nous les appelons des lieux sémantiques.

Dans notre travail, nous parlerons d'« instance de lieu » pour désigner un lieu précis. On peut aussi s'intéresser à la reconnaissance de catégories de lieux, par exemple « les couloirs », « les bureaux ». Comme nous l'avons mentionné (voir Section II.1.2.2), la plupart des études en catégorisation d'objets et de scènes s'intéressent à des catégories définies à partir de critères d'apparence. Au contraire, les catégories de lieux sont définies fonctionnellement : quelle que soit son apparence, une cuisine est un lieu dans lequel on peut cuisiner. Une catégorie de lieu est donc caractérisée par une grande variabilité d'apparence.

Une approche intuitive du problème consisterait à dresser une carte de l'environnement et segmenter les différents lieux : ainsi, la reconnaissance de lieux se ramène à un problème de localisation pour lequel des techniques classiques existent (voir Section I.1.4). Cette approche pourrait tout à fait fonctionner dans un environnement donné [1]. Cependant, il est intéressant de pouvoir reconnaître une catégorie de lieux issus de différents environnements, éventuellement dans lesquels le robot n'est jamais allé. On peut, par exemple, chercher à reconnaître les bureaux issus de différents laboratoires. Dans cette optique, nous ne pouvons pas nous appuyer sur une carte ou d'autres informations extérieures pour la reconnaissance de lieux : un lieu doit être appris et reconnu grâce à son apparence.

En résumé, nous devons utiliser l'apparence pour reconnaître des catégories basées sur des critères fonctionnels. L'hypothèse fondamentale est donc que malgré la grande diversité dans l'apparence d'une catégorie de lieux, il est possible d'extraire des informations permettant de reconnaître la catégorie.

1.2. Reconnaissance visuelle de lieux sémantiques

En principe, la reconnaissance de lieux peut utiliser des informations allothétiques provenant de n'importe quel type de capteurs : télémètres à ultrason, télémètres infrarouge, télémètres laser, caméras (simple, stéréoscopique ou panoramique).

De nombreux travaux utilisent avec succès les informations fournies par les télémètres (voir [BS02 ; MSB05 ; BKR07]). Ces informations sont de nature géométrique : distance des obstacles, espace libre autour du robot, *etc.* Cependant, ces approches se heurtent à deux limitations. Premièrement, les informations géométriques ne sont pas suffisantes pour caractériser l'apparence d'un lieu. Du fait de la portée limitée des capteurs, deux pièces relativement grandes risquent d'être perçues de manière identique. Le problème de confusion perceptuelle rend donc la discrimination problématique. Deuxièmement, les différentes instances d'une même catégorie de lieux exhibent une grande diversité de configurations géométriques (voir Section III.3.1.3). Cette grande variabilité perceptuelle entraîne des difficultés dans la tâche de catégorisation.

La classification, notamment de nouveaux lieux, requiert donc l'utilisation de caméras qui permettent de travailler dans des espaces de représentation suffisamment riches. De plus, bien que nous ne les utilisions pas, les systèmes de caméras stéréoscopiques, par exemple la Kinect® de Microsoft, permettent d'accéder aussi à des informations géométriques.

Ces raisons font que la vision est la modalité privilégiée pour la reconnaissance de lieux sémantiques bien que des approches multimodales existent (voir section III.4).

1. C'est d'ailleurs la technique utilisée pour l'étiquetage des images dans la base COLD (voir section A.1).

1.3. Intérêt pour la robotique

1.3.1. Contexte sémantique

Un lieu sémantique étant défini de manière fonctionnelle, l'accès à cette information permet de définir le contexte d'action d'un robot, ce qui présente deux avantages.

D'une part, le contexte permet d'orienter la sélection de l'action par le système : en fonction du lieu dans lequel il se trouve certaines actions peuvent être entreprises, d'autres doivent être abandonnées.

D'autre part, comme l'ont montré les travaux de Torralba et al. (voir [Tor03 ; Tor+03a]), le contexte permet de faciliter le processus de reconnaissance des objets (voir [GB10]). On parle alors d'amorçage contextuel : étant dans un lieu (*e.g.* une cuisine) on peut s'attendre à trouver certains objets (*e.g.* une casserole) et pas d'autres (*e.g.* une télévision). En plus d'indices sur la probabilité de trouver telle ou telle catégorie d'objets dans un lieu, les travaux de Torralba et al. étudient aussi l'influence du contexte sur la position et l'échelle des objets.

1.3.2. Navigation

La reconnaissance de lieux sémantiques peut être utile dans des tâches de navigation pour au moins trois raisons.

Premièrement, la reconnaissance de lieux sémantiques peut aider à résoudre le problème de fermeture de boucle (*loop-closure detection*). Dans les approches cartographiques, si la position courante est connue avec suffisamment de précision, un robot revenant dans une zone déjà cartographiée devrait l'identifier comme telle. Mais l'accumulation d'erreurs, en particulier odométriques, au cours du temps peut rendre l'estimation de la position du robot très imprécise. Il arrive couramment dans ce cas que la détection n'ait pas lieu. En plus de poser problème pour la navigation, cette erreur conduit à la création d'un nouveau nœud sur la carte, le système pensant faire face à un lieu nouveau. Les systèmes de reconnaissance de lieux sémantiques (plus précisément la reconnaissance d'instances), basés sur des informations allothétiques donc indépendantes de l'estimation de la position du robot, peuvent permettre la détection d'une fermeture de boucle.

Deuxièmement, la reconnaissance sémantique de lieux peut être utilisée pour adapter la stratégie de navigation à la nature du lieu courant. Par exemple, dans un couloir, le robot peut adopter un comportement réactif de suivi des murs alors que dans un pièce plus vaste, une stratégie de navigation topologique point-à-point serait plus adaptée.

Troisièmement, la reconnaissance de lieux sémantiques peut être utilisée pour résoudre le problème du robot kidnappé. Ce problème se pose lorsqu'un robot est déplacé par un opérateur humain. Les techniques probabilistes de SLAM sont basées sur la prédiction des modifications de l'état du monde par les actions du robot (étape 2 de l'algorithme de filtrage bayésien présenté page 19). Ces informations étant absentes lors du kidnapping, les perceptions ne sont

pas congruentes. Bien que les techniques de filtrage bayésien permettent en partie de résoudre ce problème, si le robot est déplacé dans un nouvel environnement la carte précédemment acquise ne lui sert à rien. Dans ce cas, un robot doté d'une capacité de reconnaissance sémantique pourra néanmoins commencer à agir. La reconnaissance d'instances est parfois appelée « localisation qualitative ».

1.3.3. Cartographie sémantique

Enfin, la reconnaissance de lieux est un composant essentiel d'un système de cartographie sémantique (voir Section I.1.2.3). En effet, les lieux sémantiques correspondent à la description par un être humain de son environnement courant. La possibilité d'interagir avec un robot doté des même concepts peut permettre d'améliorer la communication homme-machine.

Lors d'une conversation ordinaire, on peut demander à quelqu'un de se rendre dans le bureau de Malika pour chercher tel ou tel objet.

En conclusion, nous avons défini dans cette section le concept de lieu sémantique et l'utilité de la tâche de la reconnaissance visuelle de lieux sémantiques. La section suivante est consacrée à la position du problème et aux différences avec les domaines voisins.

2. Position du problème

Les travaux en reconnaissance de lieux sont souvent rapprochés des travaux en Simultaneous Location And Mapping (SLAM) ou en reconnaissance de scènes. Essayer de résoudre le problème par ces approches peut amener à négliger des aspects importants des lieux sémantiques. La Section III.2.1 explique les différences par rapport au SLAM. La Section III.2.2 explique les différences par rapport à la reconnaissance de scènes. Enfin, nous exposons notre approche dans la partie III.2.3.

2.1. Différences par rapport au SLAM

Comme expliqué dans la Section I.1.2, la construction et l'utilisation d'une carte topologique nécessite de reconnaître à quel nœud l'entrée courante appartient. Ce problème est souvent appelé *place recognition* qu'on traduit par reconnaissance de lieux. Comme nous l'avons dit, les télémètres fournissent une information assez pauvre. Ainsi, de nombreux travaux ont été menés dans le domaine de la localisation topologique à partir d'information visuelle.

Les premières approches se basaient sur l'utilisation de caméra omnidirectionnelles (voir [UN00 ; VLD06]). Les caméras classiques ont aussi été utilisées. [KW95 ; TS05] utilisent à la fois des télémètres et des caméras.

Cependant, dans ces systèmes, on ne s'intéresse pas en tant que tel au lieu mais on cherche simplement à extraire des caractéristiques permettant de reconnaître le nœud en question.

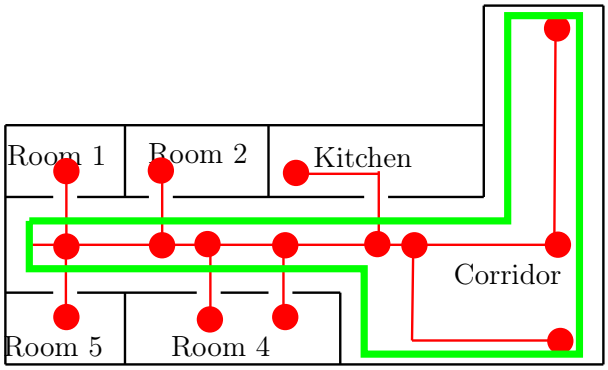


FIGURE III.1.: Différence entre le concept de lieu dans le SLAM topologique et le concept de lieu sémantique. Les points rouges représentent les nœuds d'une carte topologique, appelés lieux dans ce contexte. Les relations d'accessibilité entre ces lieux sont représentées par les traits rouges. Les nœuds groupés en vert correspondant au lieu sémantique « couloir ». Par opposition, au nœud de la carte, le couloir a une extension spatiale.

De plus, les lieux ainsi définis n'ont pas d'extension spatiale et de caractère sémantique. Ils servent juste de points de repère pour la navigation. Cette différence est illustrée dans la Figure III.1.

De le cadre de la reconnaissance de catégorie de lieux, le robot peut être confronté à des lieux jamais vus avant ou issus d'autres environnements (nous sommes capables de reconnaître un bureau même dans un laboratoire où nous ne sommes jamais allés). Les techniques de SLAM se concentrent sur la reconnaissance de lieux déjà vus : si l'entrée sensorielle n'est proche d'aucun lieu connu, un nouveau nœud est simplement ajouté. C'est pourquoi les techniques de SLAM n'abordent pas la question de la catégorisation de lieux.

2.2. Différences par rapport à la reconnaissance de scènes

De manière générale, la reconnaissance de scènes vise à identifier le contenu d'une image. On peut bien sûr présenter une image à un système de reconnaissance de scènes et lui associer comme étiquette le lieu correspondant. Cependant, un lieu ou une catégorie de lieux peuvent être caractérisés par une grande diversité de scènes (voir Section III.3.3). Comme nous le détaillerons dans la Section III.3.3, étant donné la même image d'entrée, un algorithme de reconnaissance de scènes et un algorithme de reconnaissance de lieux ne doivent pas toujours donner la même réponse. La différence essentielle vient du fait qu'une scène est définie par le contenu de l'image et n'a donc pas d'extension au-delà de l'image alors qu'un lieu sémantique est défini dans l'espace réel.

De plus, les images utilisées dans la reconnaissance de scènes sont en général prises par des humains et donc sont très représentatives de la catégorie. Les images que nous utilisons (voir Appendice A) ont été extraites de vidéos acquises par un robot. Le point de vue n'est donc pas forcément idéal. En revanche, nous disposons d'une suite d'images, acquises à une fréquence de quelques Hertz. De par la continuité du déplacement du robot, deux images successives ont une forte probabilité de se situer dans la même pièce.

2.3. Position formelle du problème

Nous disposons d'un robot qui parcourt un certain environnement et acquiert des images à l'aide d'une caméra (voir Figure III.2). La plupart du temps, le robot se trouve en intérieur. Ainsi, nous emploierons parfois le terme « pièce » pour désigner une instance, bien que certains lieux auxquels nous nous intéressons ne sont pas des pièces (les couloirs par exemple). En principe, rien n'interdit d'étudier des lieux sémantiques extérieurs comme le parking du LIMSI, la place de la République à Orsay, *etc.* Notons toutefois que les environnements extérieurs n'ont pas toujours d'unité sémantique. Néanmoins, on peut définir des catégories utiles pour une tâche donnée. Par exemple si l'on imagine un robot dont la tâche est de livrer des objets, Wu [Wu09] propose de définir des catégories telles que « rue », « trottoir », « place », *etc.*

On dispose d'un ensemble $\mathcal{I}_{\text{train}}$ de séquences vidéos pour l'entraînement. L'environnement est échantillonné régulièrement : à chaque instant t nous disposons d'une image $\mathbf{I}_t$, qui est décrite par un vecteur $\boldsymbol{x}_t$. L'apprentissage est supervisé, donc chaque observation est étiquetée en fonction du lieu où elle a été réalisée. Dans la tâche de reconnaissance d'instances, l'étiquette y_t est issue de l'ensemble discret $\mathcal{C} = \{c^1, c^2, \ldots, c^p\}$. Dans la tâche de reconnaissance de catégorie, chaque image est étiquetée avec une catégorie $\mathbf{C} = \{C^1, C^2, \ldots, C^P\}$. On dispose aussi d'un ensemble de séquences vidéos de test $\mathcal{I}_{\text{test}}$.

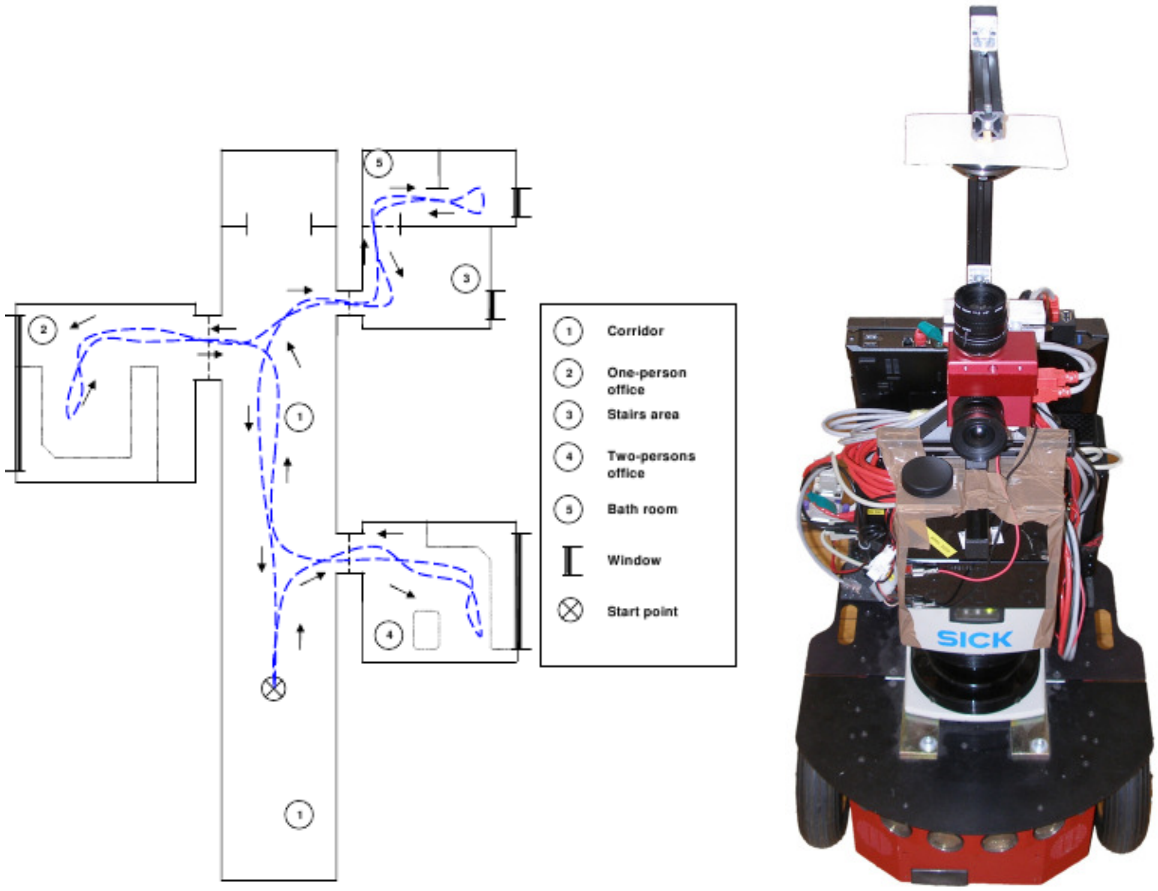
La tâche est de prédire l'étiquette y_t de l'image $\mathbf{I}_t$, prise à l'instant t, à partir de sa description $\boldsymbol{x}_t$. Éventuellement, le robot peut s'aider des images précédentes (auquel cas, on parle d'intégration temporelle). D'un point de vue probabiliste, le but est donc d'estimer $\mathrm{P}\left(y_t|\boldsymbol{x}_{1:t}\right)$.

La réponse du système est la classe qui maximise la probabilité *a posteriori*. Dans le problème de reconnaissance d'instance :

$$\overset{\star}{y}_t = \arg\max_{c^i \in \mathcal{C}} \mathrm{P}\left(y_t = c^i|\boldsymbol{x}_{1:t}\right) \tag{III.1}$$

Dans le problème de catégorisation :

$$\overset{\star}{y}_t = \arg\max_{C^i \in \mathbf{C}} \mathrm{P}\left(y_t = C^i|\boldsymbol{x}_{1:t}\right) \tag{III.2}$$



(a) Exemple d'environnement avec le parcours du robot. Issu de [Ull+07].

(b) Robot et caméra utilisés. Issu de [Ull+07].

FIGURE III.2.: Scénario pour la reconnaissance de lieux sémantiques.

Avant de passer en revue les travaux sur la reconnaissance de lieux sémantiques, nous expliquons les problèmes spécifiques soulevés par cette tâche par rapport à ceux de la reconnaissance visuelle en général et de la reconnaissance de scènes en particulier que nous avons exposés au chapitre précédent. Dans la suite du manuscrit, si le contexte n'est pas ambigu, lorsque nous parlons de « lieux », il faut entendre « lieux sémantiques ».

3. Challenges

La reconnaissance de lieux sémantiques présente des difficultés particulières dont l'origine se trouve à la fois dans la définition du problème et dans l'utilisation d'un système robotique. Nous proposons de classer ces difficultés en trois grands groupes. Tout d'abord, les lieux sont susceptibles d'une forte variabilité, ce qui complique la reconnaissance d'instances et la catégorisation. Nous présentons les sources de variabilité dans la Section III.3.1. Ensuite, l'utilisation d'une plate-forme robotique génère un certain nombre de problèmes dans la prise de vue des images (voir Section III.3.2). Enfin, la Section III.3.3 présente les problèmes liés à l'extension spatiale d'un lieu.

3.1. Variabilité des lieux

3.1.1. Variations dynamiques

L'apparence d'un lieu se modifie au cours du temps principalement à cause de 2 facteurs : les variations d'illumination et l'activité des occupants.

L'apparence d'une pièce varie grandement en fonction de l'éclairage ambiant, par exemple entre le jour et la nuit ou bien en présence de lumière artificielle. Les environnements d'intérieur possèdent généralement des fenêtres. Ainsi, leur apparence dépend des conditions météorologiques (voir Figure III.3). Par exemple, un jour de soleil va provoquer des réflexions sur les surfaces et des ombres qui seront absentes un jour pluvieux (mais les utilisateurs du bâtiment peuvent tirer les rideaux). De même, la nuit une pièce non-éclairée sera très sombre. L'éclairage artificiel a des propriétés sensiblement différentes de la lumière du soleil (spectre, incidence). De plus, la plupart des caméras disposent d'une balance automatique des blancs et d'autres mécanismes d'amélioration de l'apparence des images. Lors du déplacement du robot, les conditions de prise de vue peuvent changer (par exemple, suivant que l'on est devant une fenêtre ou qu'on y tourne le dos), ce qui produit des changements dans la balance des blancs et donc dans la répartition des couleurs. La robustesse des méthodes de reconnaissance de lieux à ces variations est un enjeu important.

Une autre source de variation est liée au fait que les pièces sont utilisées par des êtres humains : les occupants d'une pièce sont amenés à en modifier l'apparence de par les diverses activités qu'ils peuvent y mener. Ces modifications peuvent être temporaires ou occuper des

(a) Jour nuageux (b) Jour ensoleillé (c) Nuit

FIGURE III.3.: Exemple de l'influence de la luminosité ambiante sur l'apparence d'un même lieu. Issu de [Ull+07].

échelles de temps relativement importantes (voir Figure III.4). Un exemple de modification temporaire correspond au déplacement d'objets dans la pièce, par exemple une chaise. De plus, des personnes (ou d'autres robots) peuvent traverser la pièce et donc modifier temporairement son apparence. L'apparence peut aussi se modifier graduellement (par exemple, les articles s'accumulant sur le bureau d'un thésard en période de rédaction) ou au contraire rapidement (par exemple lors d'un grand ménage ou d'une réorganisation de la pièce).

Ces variations dynamiques, intrinsèques à la définition d'un lieu, rendent difficile la reconnaissance d'un même lieu lorsqu'il est visité à plusieurs reprises.

3.1.2. Ressemblance des pièces dans un environnement donné

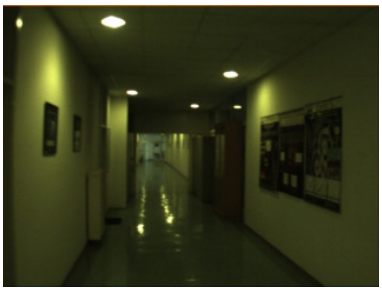
Dans un laboratoire, la plupart des bureaux se ressemblent : on va retrouver les mêmes couleurs dominantes, les mêmes volumes (voir Figure III.5).

Dans une tâche de reconnaissance d'instances, cette faible variabilité à l'intérieur d'un environnement donné pose problème pour la discrimination entre lieux appartenant à la même catégorie fonctionnelle : par exemple, il est difficile de discriminer deux bureaux de la Figure III.5.

3.1.3. Variabilité intra-catégorie

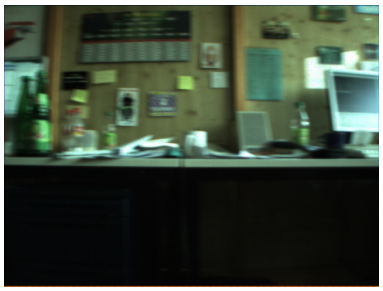
En termes d'apparence, un bureau dans un laboratoire donné (le LIMSI par exemple) sera probablement différent d'un bureau dans un autre laboratoire. Le problème de la grande variabilité intra-classe (voir Figure III.6) est donc particulièrement aigu dans une tâche de catégorisation sur plusieurs environnements (cette source de variabilité est parfois appelée « variabilité environnementale »). De plus, il est possible qu'une méthode fonctionne bien

(a) Présence de personnes



(b) Déplacement d'objets

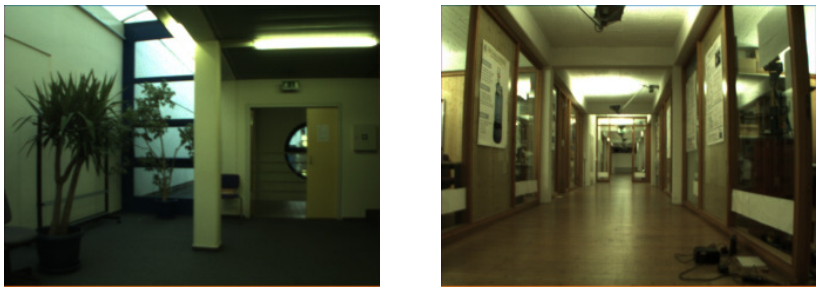
FIGURE III.4.: Exemple de l'influence de l'activité des occupants sur l'apparence d'un lieu. Issu de [Ull+07].



(a)

(b)

FIGURE III.5.: Exemple des ressemblances entre des pièces de la même catégorie dans un environnement donné. Issu de [Ull+07].



(a) (b)

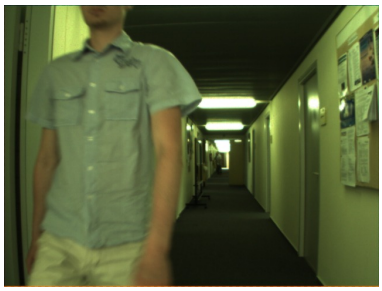
FIGURE III.6.: Exemples de variabilité entre des pièces issues de la même catégorie fonctionnelle (des couloirs). Issu de [Ull+07].

(a) (b) (c)

FIGURE III.7.: Les objets (ici les chaises) exhibent des configurations spatiales très différentes dans des lieux d'une même catégorie (ici des bureaux). Issu de [Gui09].

dans un environnement donné mais fonctionne mal dans un autre.

Cette variabilité d'apparence se retrouve notamment dans la variabilité de la disposition des objets. Comme nous l'avons vu (voir Section II.1.2), les objets appartenant à une même catégorie tendent à conserver une organisation spatiale stable de traits caractéristiques (un visage présente toujours la même configuration d'éléments tels que les yeux, le nez et la bouche). Au contraire, au sein d'une même catégorie de lieu, la configuration spatiale des objets peut varier très fortement (voir Figure III.7). Dans ces conditions, il est difficile de prédire la configuration spatiale des chaises et des tables dans un bureau.



(a) Personne masquant une partie importante du lieu.

(b) Image obtenue lors d'une manœuvre.

FIGURE III.8.: Exemples d'occultation du lieu. Issu de [Ull+07].

3.2. Difficultés liées aux images

3.2.1. Occultation

Le problème d'occultation peut se produire, par exemple lorsqu'une personne passe devant la caméra du robot et occulte partiellement sa vue (voir Figure III.8). Un autre cas d'occultation peut se produire lorsque le robot manœuvre et se trouve face à un objet ou un pan de mur (voir section suivante).

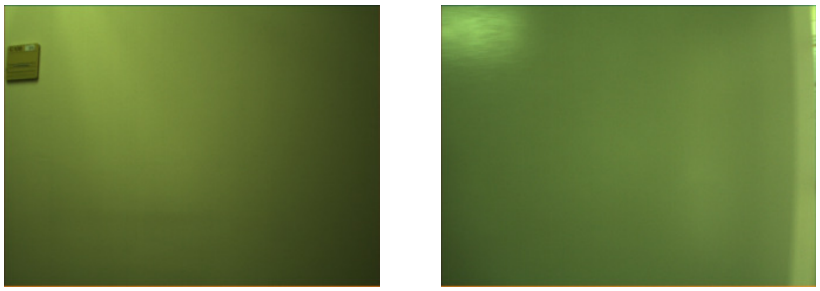
Notons que ce problème n'a pas d'équivalent dans le cadre classique de la classification de scènes, où l'on ne cherche qu'à étiqueter ce que l'on voit et non l'endroit où l'on se trouve. Rappelons qu'un objet ou un lieu sont définis dans l'espace physique. Par conséquent ils sont caractérisés par des attributs intrinsèques, qui leur appartiennent en propre et leurs attributs peuvent être partiellement masqués, auquel cas leur représentation sera modifiée. À l'inverse, une scène visuelle n'a pas d'autre extension que celle imposée par les limites de l'image. Par conséquent, tant que l'on ne cherche pas à décrire l'image en termes d'objets, il n'est pas possible de parler d'occultation [2].

Le phénomène d'occultation d'un lieu est bien sûr temporaire. Nous verrons, que cette difficulté peut être surmontée si on intègre la perception du robot dans le temps en travaillant à partir du flux vidéo.

3.2.2. Qualité des images

En robotique les images sont issues d'une séquence vidéo obtenue lors du déplacement d'un robot dans un environnement. De plus, le robot est en général guidé manuellement pendant cette opération. On est donc susceptible de rencontrer deux problèmes :

2. Le lecteur peut se reporter à la Section II.1.3.2 pour une discussion concernant l'occultation dans les scènes.



(a) (b)

FIGURE III.9.: Exemples d'images portant peu d'information. Issu de [Ull+07].

1. Images de faible qualité : le robot étant en mouvement, certaines images sont floues ; un autre problème vient des images surexposées, par exemple lorsque le robot fait face à une fenêtre.
2. Images de faible intérêt : lorsque que le robot manœuvre il se retrouve parfois proche des murs ou de certains objets ; les images obtenues sont souvent assez uniformes et sans intérêt (voir Figure III.9) [3] ; de plus, si la plupart des auteurs utilisent des caméras montées à hauteur d'homme, dans certaines bases de données le robot est situé près du sol, ce qui introduit un biais important (réflexions sur le sol, faible champ de vision du à la présence d'objets sur le sol ou de meubles).

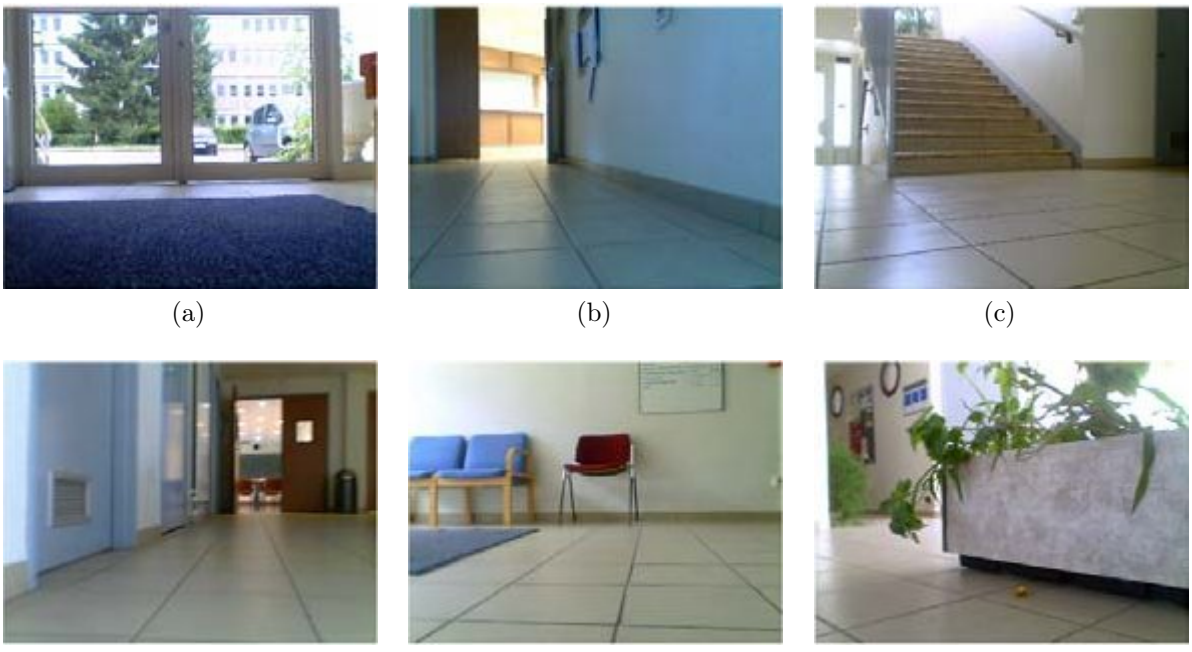
Notons que TORRALBA et al. [Tor+03b] utilisent une base de données obtenue en plaçant une caméra sur la tête de sujets humains. On peut donc supposer qu'il y a moins d'images de faible intérêt (les mécanismes d'attention visuelle portant le regard vers des zones informatives).

3.3. Extension spatiale d'un lieu et continuité temporelle

3.3.1. Modifications du point de vue

Comme nous l'avons dit, un lieu sémantique a une extension spatiale. Or le robot a un champ perceptif limité. Il ne peut donc pas percevoir l'intégralité de la pièce. Même en utilisant une caméra à 360°, il se peut que la géométrie de la pièce l'en empêche (par exemple, un couloir avec des tournants). Ainsi, selon le point de vue du robot, un même lieu peut contenir des scènes très différentes.

3. Ce jugement est à nuancer : il peut s'avérer au contraire qu'une vue particulière soit très caractéristique d'un lieu. Par exemple, les murs d'une salle de bain recouverts de carrelage forment un motif facilement reconnaissable et très discriminant. Néanmoins, ce genre de cas est probablement rare, surtout dans une tâche de catégorisation.



(a) (b) (c)

(d) (e) (f)

FIGURE III.10.: Différents vues d'un même lieu peuvent exhiber de grandes différences. Issu de [Gui09].

Les différentes vues d'un lieu ne couvrent pas les mêmes sections de l'espace et donc ne partagent pas ou peu de traits communs (voir Figure III.10). Lorsque l'on cherche à caractériser un lieu, ces scènes disjointes sont problématiques.

3.3.2. Vues ambiguës

De plus, certaines vues d'un lieu sont ambiguës ou même trompeuses pour la classification. Ce phénomène se produit notamment avec des vitres : dans ce cas, le robot peut entrevoir une autre pièce ou l'extérieur (voir figure III.11a). Dans le cas de la figure III.11a, un algorithme de reconnaissance de scènes pourrait répondre « extérieur » alors qu'un algorithme de reconnaissance de lieux répondrait « couloir ». Ce phénomène illustre bien la différence fondamentale entre la reconnaissance de lieux et la reconnaissance de scènes.

3.3.3. Transitions entre lieux

Un problème propre à la reconnaissance de lieux vient du fait que le passage d'un lieu à un autre n'est pas toujours bien défini. Dans la plupart des cas, une porte marque la frontière.

(a) Observation de l'extérieur à travers une surface transparente.

(b) Observation d'un autre lieu à l'approche des transitions.

FIGURE III.11.: Différences entre la reconnaissance de scènes et la reconnaissance de lieux sémantiques. En (a), le robot est dans un couloir mais observe l'extérieur. En (b), le robot est dans le couloir mais étant proche d'une transition, il observe une autre pièce.

Cependant, si on imagine un robot en train de passer d'un couloir à un bureau, à l'approche de la séparation, une partie importante de l'image va être occupée par une vue du bureau, alors que le robot est toujours dans le couloir. Par exemple, dans le cas de la figure III.11b, la réponse attendue du système est « couloir » et non « bureau ». Le fait que l'on puisse observer un lieu depuis un autre est une des différences fondamentales entre la reconnaissance de lieux et la reconnaissance de scènes.

Dans certaines bases de données (par exemple la base VPC décrite dans la section A.2), il existe une catégorie spéciale utilisée pour annoter les transitions entre lieux.

3.4. Conclusion

En conclusion, nous pouvons dire que la tâche de reconnaissance de lieux possède des difficultés importantes liées à la fois à la nature du problème (extension spatiale) et à l'origine des données (un robot en déplacement). Tout d'abord, du point de vue de l'apprentissage, la tâche de reconnaissance d'instances est caractérisée par une large variance intra-classe et une faible variance inter-classes. La tâche de catégorisation est caractérisée par une très large variance intra-classe. Ces difficultés ont entraîné la création d'approches spécifiques que nous détaillons dans la section suivante.

4. État de l'art en reconnaissance de lieux sémantiques

On peut distinguer plusieurs sources d'influence sur les travaux en reconnaissance de lieux :

1. L'influence du SLAM et de la localisation topologique dans laquelle on cherche à reconnaître à quel nœud l'image courante est associée.
2. Les techniques de recherche d'images qui vont rechercher dans une base de données l'image la plus proche de l'image courante pour identifier le lieu.
3. Les techniques de classification de scènes dans lesquelles on classe l'image courante en décrivant éventuellement les objets présents.

Nous présenterons tout d'abord les meilleures performances en reconnaissance d'instances et en catégorisation (Section III.4.1) reportées dans la littérature. Nous aborderons ensuite plus spécifiquement la question des formats de description de l'image (voir Section III.4.2). Enfin, nous détaillerons les techniques d'apprentissage mises en jeu (Section III.4.3) et les mécanismes utilisés pour l'intégration temporelle (Section III.4.4).

4.1. Vue d'ensemble de l'état de l'art

4.1.1. Reconnaissance d'instances

Comme nous l'avons dit, de nombreux travaux en reconnaissance d'instances abordent le problème sous l'angle de la localisation topologique.

Les travaux dans ce domaine se poursuivent, notamment pour la cartographie de grands environnements en extérieur. Les travaux de Cummins et Newman (voir [CN08 ; CN11]) et Milford et Wyeth [MW08] sont représentatifs de ces approches et permettent de cartographier des environnements de plusieurs centaines de kilomètres. Un autre axe de travail concerne l'apprentissage des variations saisonnières des environnements extérieurs (voir [Ora+07 ; VL10]).

Un certain nombre de travaux (voir [WBB02 ; FEN07]) utilisent des techniques d'indexation d'images. La reconnaissance du lieu courant est effectuée en recherchant dans la base de données l'image la plus proche.

Les travaux issus de la catégorisation de scènes ou de l'indexation d'images n'utilisent pas, en général, de méthode d'intégration temporelle. Par exemple, Ullah et al. [Ull+08] utilisent le descripteur Scale-Invariant Feature Transform (SIFT) pour décrire les images. Chaque image est classée indépendamment grâce à des Support Vector Machines.

De manière générale, les meilleures méthodes atteignent des taux de classification correcte assez élevés, parfois supérieurs à 90%. Cependant, la plupart de ces méthodes ne testent pas la robustesse de la classification. Pronobis et Caputo [PC07] utilisent la base IDOL qui permet de tester la robustesse aux changements de condition d'illumination. Les taux de reconnaissance atteignent 95% quand les conditions d'illumination sont les mêmes en

apprentissage et en reconnaissance mais chutent à environ 75% quand ce n'est pas le cas. Ullah et al. [Ull+08] utilisent la base COLD (voir section A.1) pour tester de manière plus complète la robustesse aux changements dynamiques et à la variabilité environnementale. Une partie des résultats est montrée dans la Figure III.12 (le protocole suivi est expliqué dans la Section A.1.4). Le taux de classification correcte atteint en moyenne 83,70% quand les conditions d'illumination sont les mêmes en apprentissage et en reconnaissance (les résultats sont du même ordre dans les autres laboratoires). Le système est assez robuste aux changements de conditions d'illumination (voir Figure III.12).

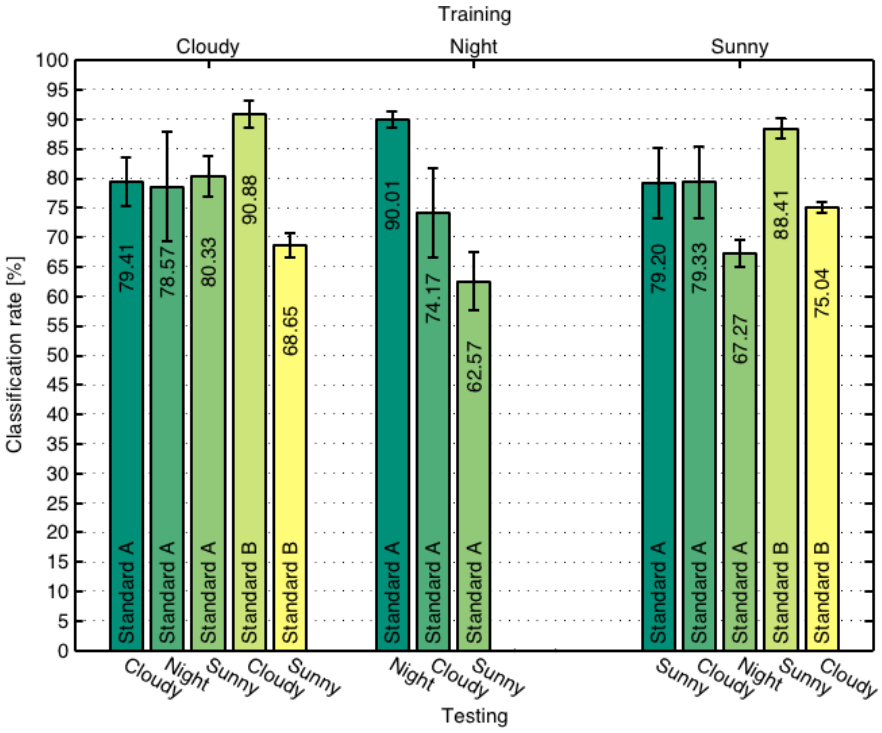


Figure III.12.: Résultats en reconnaissance d'instances obtenus par Ullah et al. [Ull+08] sur la base COLD. Les résultats ont été obtenus dans le laboratoire de Fribourg avec les séquences standard. L'apprentissage a été effectué dans une condition (indiquée sur l'axe horizontal supérieur) et la reconnaissance dans la même condition ou une autre (indiquée sur l'axe horizontal inférieur). On peut voir sur ce graphique que le système est assez robuste aux changements de conditions d'illumination : pour les trois barres de gauche l'apprentissage a été effectué dans la partie A du laboratoire avec les séquences nuageuses et la reconnaissance avec les séquences nuageuses, de nuit et ensoleillées respectivement ; les performances sont à peu près identiques. Tiré de [Ull+08].

Enfin, les méthodes multimodales, par exemple le système développé par [Pro+10] qui combine l'utilisation de caméras et de télémètres lasers, offrent en général de bonnes performances et une bonne robustesse aux conditions d'illumination (les télémètres lasers étant insensibles aux variations de luminosité ambiante).

4.1.2. Catégorisation

La tâche de catégorisation a d'abord été étudiée dans le domaine de la vision par ordinateur, en lien avec les travaux en catégorisation de scènes dont nous avons parlé dans le chapitre II (voir page 69). Torralba et al. [Tor+03b] est l'un des premiers travaux à traiter le problème de la catégorisation de lieux sémantiques. L'approche utilise le descripteur GIST et a été testée en reconnaissance d'instances et en catégorisation, sur des lieux intérieurs et extérieurs. L'intégration temporelle est basée sur un Modèle de Markov Caché (MMC) [4] pour fusionner des informations au fil du temps et de l'espace. Les taux de classification correcte sont de l'ordre de 40% sur 10 catégories (mais les résultats ont été obtenus sur une seule séquence donc ne prennent pas en compte la variabilité dynamique).

Wu, Christensen et Rehg [WCR09] utilisent le descripteur global CENTRIST décrit dans la Section I.2.2.2. Là aussi, un HMM est utilisé pour intégrer temporellement les entrées. Le système est évalué sur cinq catégories issues de la base VPC (voir section A.2 ; le protocole est expliqué dans la Section A.2.3) et montre des taux de catégorisation correcte allant de 20,80% pour les salles à manger à 74,34% pour les salles de bain (voir Table III.1). Cependant, le taux de classification correcte par catégorie montre aussi de fortes variations (atteignant parfois 0% pour certaines catégories) et il n'y a pas de tests de robustesse aux changements dynamiques.

Ranganathan [Ran10] utilise une modélisation explicite des transitions entre lieux, appelée *change point detection* : les transitons sont détectées par des changements abrupts dans la statistique du signal. Le système a été évalué de la même manière que [WCR09] : les résultats sont légèrement moins bons que ceux de Wu (voir Table III.1).

Ullah et al. [Ull+08] testent leur système sur quatre catégories issues de la base COLD. Le système est entraîné avec des séquences issues de deux laboratoires et la reconnaissance s'effectue dans le troisième (le protocole est décrit Section A.1.5). Les conditions de luminosité sont les mêmes pour les séquences d'apprentissage et les séquences d'entraînement (on teste la sensibilité aux variations dynamiques mais pas aux conditions d'éclairement). Les taux de classification correcte sont au-dessous de 25% sauf pour le couloir (voir Figure III.13) et montrent une forte sensibilité aux changements dynamiques.

Cette partie a présenté les systèmes les plus performants dans le domaine. La partie suivante passe en revue les différents formats de représentation des images utilisés dans la littérature.

4. Nous désignerons le plus souvent ces techniques par leur nom anglais : *Hidden Markov Model (HMM)*.

	Bedroom	Bathroom	Kitchen	Living room	Dining room	Transition	Average
[Wu09] CENTRIST	66,44%	74,34%	44,40%	27,89%	20,80%		46,78%
[Ran10] CENTRIST	61,81%	65,68%	48,27%	24,07%	16,18%	37,43%	42,24%
[Ran10] SIFT	61,72%	69,61%	51,11%	29,08%	14,94%	42,84%	44,88%

TABLE III.1.: Résultats en catégorisation obtenus par [Wu09] et [Ran10]. L'apprentissage est réalisé sur 5 maisons et testé sur la maison restante. Le tableau montre le taux de catégorisation correcte moyenne pour chaque catégorie. Il fut noter que certaines catégories sont très bien reconnues dans un environnement donné et très mal dans d'autres (par exemple). Adapté de [Wu09 ; Ran10].

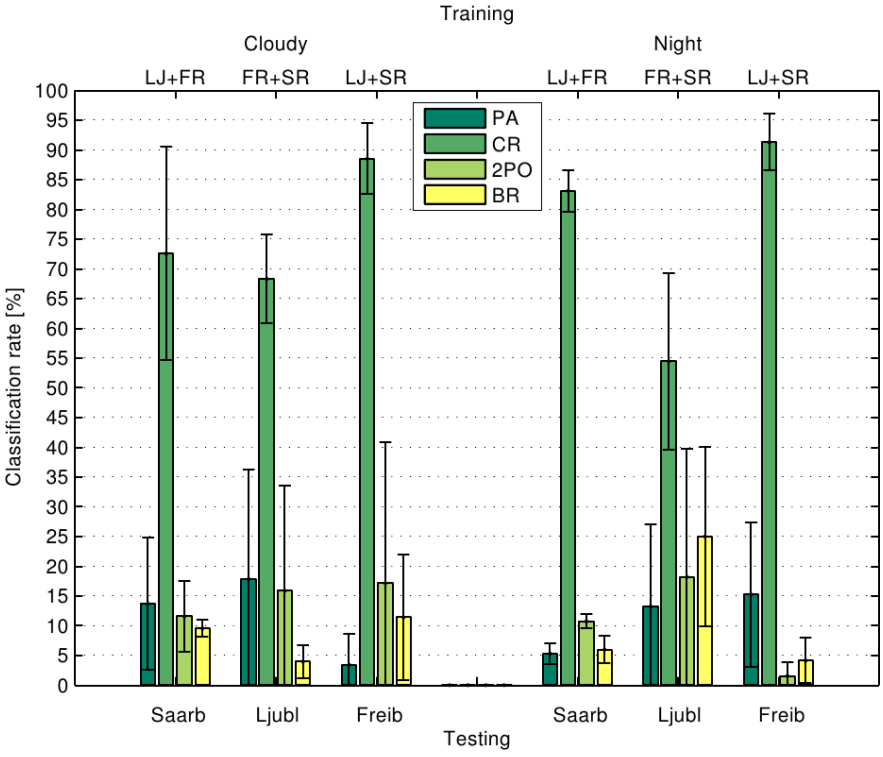


FIGURE III.13.: Résultats en catégorisation obtenus par ULLAH et al. [Ull+08] sur la base COLD. Tiré de [Ull+08].

4.2. Formats de description de l'image

Nous détaillons ici les techniques de description de l'image utilisées en reconnaissance de lieux. Nous retrouvons les grandes distinctions que nous avons pu faire pour la reconnaissance de scènes entre les approches bas-niveau, les approches basées sur les mots visuels et les approches basées sur les objets (approches sémantiques).

4.2.1. Approches bas-niveau

Description locale La plupart des systèmes se basent sur le descripteur SIFT ([KF04 ; Ora+07]), ou des descripteurs proches de celui-ci ([VL10 ; CN08]), qui offrent les meilleures performances.

En ce qui concerne la classification de scènes, nous avons vu que ces méthodes sont en général très efficaces en reconnaissance d'instances mais moins en catégorisation.

Description globale Les techniques employées sont identiques à celles utilisées pour la représentation des scènes à partir de caractéristiques globales (cf Section II.2.1). La plupart des travaux qui utilisent ce genre de codages sont orienté vers la catégorisation des scènes. Les traits les plus utilisés sont les couleurs et les orientations (voir [GO00]). Les informations de couleur semblent utiles pour la reconnaissance d'instances mais ont de mauvaises performances en catégorisation ([Tor+03b]).

Un certain nombre de travaux utilisent à la fois une description globale et locale (voir [PC07 ; QT09]), ce qui permet d'augmenter les performances.

4.2.2. Approches basées sur les mots visuels

Un certain nombre d'auteurs ont utilisé les méthodes Bag-of-Words (BoW) (voir [Fil08 ; Ran10]), généralement avec de bonnes performances.

Wu, Christensen et Rehg [WCR09] combinent une signature globale localisée, CENTRIST, avec les méthodes BoW (bien qu'il ne fasse pas le lien de manière explicite). Nous reviendrons sur ce point dans le chapitre VI car ce travail est très proche du nôtre.

4.2.3. Description basée sur les objets

Certains travaux s'inspirent des modèles de la reconnaissance d'objets pour reconnaître les lieux : dans un premier temps, les objets sont d'abord reconnus puis cette information est utilisée pour reconnaître le lieu. Ces travaux ont été utilisés en reconnaissance d'instances (voir [RD07 ; VS07 ; Moz+06]) et en catégorisation (voir [Vis+10]).

Les travaux de Mozos et al. [Moz+06] utilisent un petit nombre d'instances d'objets (en plus d'informations fournies par des capteurs laser) pour reconnaître les lieux.

Viswanathan et al. [Vis+10] adoptent une approche purement basée sur les objets : les lieux sont décrits grâce aux objets qui s'y trouvent. Les modèles d'objets utilisent les méthodes vues à la Section II.4.2.2. L'apprentissage nécessite une large base d'apprentissage d'objets segmentés (la base LabelMe). Le modèle atteint un taux de classification correcte de l'ordre de 75% mais sur une base de données de 67 images contenant seulement deux catégories (« cuisine » et « bureau »).

Ranganathan et Dellaert [RD07] utilisent un modèle de constellation étendu en 3D pour décrire l'apparence et la localisation des objets dans un lieu. Ce travail n'a été utilisé qu'en reconnaissance d'instances.

Vasudevan et Siegwart [VS07] proposent une représentation hiérarchique de l'espace basée sur les objets : les objets sont d'abord regroupés en catégories puis ces catégories sont utilisées pour construire le modèle de lieu (éventuellement en utilisant la distance entre les objets).

Ces deux travaux emploient des caméras stéréoscopiques, en combinaison avec des lasers pour détecter les portes. En revanche, ces travaux n'emploient pas, en général, de mécanisme d'intégration temporelle.

4.2.4. Conclusion

Globalement, les méthodes basées sur les objets nécessitent des modèles complexes et des procédures d'apprentissage lourdes. De plus, leur utilisation en catégorisation n'est pas triviale. Au final, ces systèmes tombent plus dans le cadre de la cartographie sémantique, abordé Section I.1.2.3, que dans celui de la reconnaissance de lieux proprement dite.

En tout état de cause, au vu du nombre restreint de travaux dans ce domaine, il est difficile de conclure si un codage global ou local est le plus adapté. Les codages globaux semblent toutefois avoir de bonnes performances. En ce qui concerne les indices visuels utilisés, si les couleurs semblent intéressantes pour la reconnaissance d'instances, ce n'est pas le cas pour la catégorisation.

4.3. Classification et confiance dans la décision

Nous détaillons ici les techniques d'apprentissage utilisées en reconnaissance de lieux. Comme nous l'avons dit (voir Section I.3.2), on distingue deux grandes familles de méthodes d'apprentissage dont nous avons déjà discuté les avantages et les inconvénients : les méthodes génératives et les méthodes discriminatives.

4.3.1. Techniques d'apprentissage utilisées

La plupart des travaux utilisent un apprentissage discriminatif de type SVM ([PC07 ; Ull+08]) car celui-ci offre de bonnes performances. Certains utilisent un apprentissage

génératif ([Tor+03b ; VS07 ; Wu09 ; Ran10]). Au niveau des performances, les deux approches semblent fournir de bons résultats mais seul le système de [Ull+08] a été évalué en profondeur.

Dans les méthodes BoW, la phase de quantification vectorielle est, en général, hors-ligne et non-incrémentale. Une exception est constituée par les travaux de [Fil08], qui présentent un algorithme entièrement incrémental (algorithme de classification non-supervisée et algorithme d'apprentissage) et interactif, c'est-à-dire que l'utilisateur du système peut corriger celui-ci quand il se trompe. Le robot est ainsi fortement supervisé en début d'apprentissage, lorsqu'il commet le plus d'erreurs, puis le niveau de supervision diminue.

4.3.2. Mesure de la confiance

En général, un système de reconnaissance de lieux doit donner une réponse pour chaque image. Cependant, nous avons souligné (voir Section III.3.3) que certaines images sont ambiguës pour la reconnaissance de lieux. Dès lors, il est intéressant de quantifier la confiance qu'on a dans la classification d'une image. En effet, si celle-ci est faible, le robot peut chercher à collecter d'autres indices pour améliorer la classification.

Un des avantages des méthodes génératives est qu'il est relativement facile de quantifier l'incertitude du système. En effet, dans ces méthodes, on modélise la vraisemblance de l'observation $\mathrm{p}(\boldsymbol{x}|y)$. On peut dès lors utiliser un test statistique (appelé *likelihood-ratio test*) afin de déterminer si l'observation est correctement expliquée par un modèle de lieu (voir [Ran10]). Si le test est négatif pour chacun des modèles, on peut en déduire que l'observation n'est pas bien expliquée par l'état des connaissances.

Les méthodes discriminatives ne permettent pas, en général, de quantifier l'incertitude de la classification. Cependant, les travaux de Pronobis et Caputo [PC07] ont introduit une mesure empirique de confiance pour les SVM dans le cas d'un problème multiclasse *one-against-all* (voir page 52). Rappelons que dans ce cas, on calcule p hyperplans, chacun séparant une classe de toutes les autres. Lorsqu'on présente, un exemple $\boldsymbol{x}$, on calcule un score pour chaque classe

$$\mathrm{V}_i(\boldsymbol{x}) = \left|\mathrm{D}_i(\boldsymbol{x}) - \bar{D}_i\right|$$

où :

- $\mathrm{D}_i(\boldsymbol{x})$ représente la distance entre l'observation $\boldsymbol{x}$ et l'hyperplan h_i séparant la classe c^i des autres classes.
- $\bar{D}_i$ représente la distance moyenne à l'hyperplan h_i des exemples de la classe c^i.

Ce score représente la proximité entre $\boldsymbol{x}$ et les données de la classe c^i (qui ont servi à déterminer h_i). La classe retenue est celle qui a le score le moins élevé : $\widehat{y} = \arg\min_{c^i \in \mathcal{C}} \mathrm{V}_i(\boldsymbol{x})$ et la confiance retenue est la différence entre les deux meilleurs scores :

$$\mathrm{C}(\boldsymbol{x}) = \min_{c^i \in \mathcal{C}, c^i \neq \widehat{y}} \mathrm{V}_j(\boldsymbol{x}) - \mathrm{V}_{\widehat{y}}(\boldsymbol{x})$$

Filliat [Fil08] utilise une technique proche pour les systèmes à base de vote. Dans de tels systèmes, on dispose de plusieurs classificateurs, chacun étant entraîné sur un type de caractéristiques particulier (*e.g.* SIFT, histogramme de la teinte, *etc.*). En phase de reconnaissance, chaque classificateur vote pour une catégorie et la catégorie gagnante est celle qui a le plus de voix. La qualité du vote est calculée comme suit :

$$q = \frac{n_{winner} - n_{second}}{\sum_{c^i \in \mathcal{C}} n_i}$$

où n_{winner} désigne le nombre de votes de la classe gagnant et n_{second} le nombre de votes de la classe arrivée seconde.

Les systèmes basés sur les techniques de recherche d'image (voir [WBB02 ; FEN07]) fournissent une mesure pratique de la confiance du système basée sur la distance entre l'image trouvée et l'image requête.

4.3.3. Augmentation de la confiance

On peut distinguer plusieurs approches pour réagir en cas de classification peu concluante. Dans [PC07], le système utilise d'abord une signature globale pour classer l'image. Si la confiance n'est pas suffisante, il utilise alors une classification à partir d'une signature locale, puis les classifications sont combinées (en pondérant par la confiance). [Fil08] propose un système actif : en cas de doute, le robot tourne la tête pour acquérir de nouvelles vues.

En conclusion, l'utilisation par le système de la confiance en sa classification permet d'augmenter la qualité de celle-ci. Bien que cela soit possible avec des approches discriminatives, ces règles sont moins bien fondées que dans l'approche générative.

4.4. Intégration temporelle

Un lieu sémantique est caractérisé par une certaine étendue spatiale. De plus, la vitesse de déplacement du robot est faible. Il est donc probable que le robot reste un certain temps dans la même pièce.

En outre, en se déplaçant, le robot va générer une séquence d'entrées sensorielles corrélées. Il semble donc naturel de tenir compte de plusieurs images pour améliorer la continuité temporelle de la classification.

L'intégration temporelle désigne les techniques utilisées pour prendre en compte plusieurs images. De manière générale, l'utilisation de plusieurs images permet de désambiguïser la classification mais pose le problème de la détection des transitions entre lieux.

4.4.1. Filtrage bayésien

La technique d'intégration temporelle la plus utilisée est sans doute le filtrage bayésien (ou HMM) dont nous avons parlé dans la Section I.1.4 (voir par exemple [Tor+03b ; Moz+07 ;

KF04 ; WCR09]).

Dans ce cas, l'état caché à estimer, $\boldsymbol{Y}_t$, est le lieu (ou la catégorie de lieu) à l'instant t. C'est en général une variable aléatoire discrète. L'observation $\boldsymbol{X}_t$ est la représentation de l'image. En général, on ne modélise pas les actions car le robot est guidé manuellement (de plus, les images étant échantillonnées régulièrement, le déplacement entre deux images est à peu près constant). Ainsi, pour utiliser le filtrage bayésien, on doit spécifier les distributions de probabilité suivantes :

- La densité $\mathrm{p}\left(\boldsymbol{X}_t|\boldsymbol{Y}_t\right)$ qui est donnée par l'apprentissage du modèle de lieu dans un cadre génératif. La complexité de cette distribution est fonction du mode de représentation des images : dans un modèle basé sur les objets, elle modélise l'apparence et éventuellement la position des objets en fonction du lieu ([RD07]) ; dans un modèle basé sur une représentation globale, $\boldsymbol{X}_t$ représente la signature globale de l'image ; dans un modèle basé sur les caractéristiques locales, elle peut prendre en compte la position, l'apparence et la position des points-clés. L'utilisation d'une étape de discrétisation permet de réduire la complexité puisque dans ce cas, on doit estimer une distribution discrète de probabilité et non une densité.
- La probabilité de changement d'état $\mathrm{P}\left(\boldsymbol{Y}_t|\boldsymbol{Y}_{t-1}\right)$. Cette distribution de probabilité représente la probabilité de passer d'un lieu à un autre (ou d'une catégorie à une autre) entre l'image prise à l'instant $t-1$ et l'image prise à l'instant t. Elle est donc représentative de la manière dont les pièces sont connectées (la « topologie » de l'environnement) : on peut donc soit l'estimer à partir des données d'apprentissage (voir [Tor+03b ; Moz+07]), soit considérer qu'elle est pré-définie (voir [WCR09]). Dans le premier cas, la distribution de probabilité est spécifique à l'environnement en question, ce qui peut poser des problèmes si l'on essaye de reconnaître dans un autre environnement. Dans le second cas, on spécifie une probabilité assez forte de rester dans le même lieu (la fréquence d'acquisition des images étant en général de quelques Hertz). Notons que certaines bases de données définissent une catégorie pour les zones intermédiaires entre deux lieux (par exemple la catégorie « transition » dans la base VPC ou la catégorie « doorway » dans les travaux de Mozos). Dans ce cas, la probabilité de transition d'une pièce à une autre doit en tenir compte.

Notons que le filtrage bayésien ne nécessite pas de détecter le changement de lieu : celui-ci est prédit sur la base de la suite des observations et des classifications antérieures. L'estimation du lieu courant s'appuie sur toute la trajectoire du robot. Dès lors, une exploration selon un chemin différent peut conduire à de mauvaises performances.

[Tor+03b] rapporte des gains de 20% sur une tâche de catégorisation avec l'utilisation du filtrage bayésien. [WCR09] rapporte des gains plus faibles et dépendants de la catégorie : pour certaines catégories, le taux de reconnaissance augmente de 10% alors que dans certains cas, le taux de reconnaissance diminue.

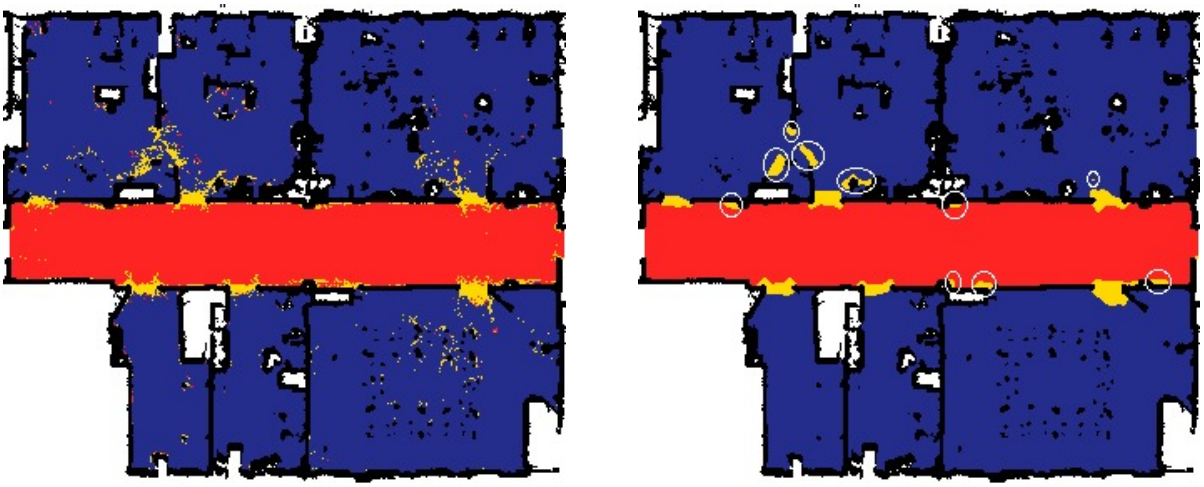
4.4.2. Autres techniques

Approche spatiale D'autres approches de l'intégration temporelle sont basées sur l'utilisation d'une carte métrique de l'environnement sous la forme d'une grille d'occupation (voir figure III.14a). Dans un premier temps, on utilise une technique d'apprentissage supervisé pour associer chaque cellule de la carte à un lieu. On obtient un étiquetage semblable à celui de la carte de gauche dans la figure III.14a dans lequel certaines cellules sont mal étiquetées. On utilise ensuite des techniques de « relaxation de l'étiquetage » afin de corriger ces erreurs. L'idée générale est la suivante : si une cellule étiquetée « bureau » est entourée de cellules étiquetées « couloir », elle est probablement mal classée. On obtient ainsi une carte plus homogène (voir carte de droite dans la figure III.14a). Cet étiquetage peut être utilisé lors de la reconnaissance pour assurer l'intégration temporelle. C'est donc une approche spatiale de la continuité temporelle. Les techniques employées sont assez lourdes : *conditional random fields* (voir [FPF07]) ou champ de Markov associatifs (voir [Moz+06]). De plus, comme toute approche cartographique (voir Section III.2.1), la carte obtenue est propre à un environnement et ne permet pas la généralisation à de nouveaux environnements.

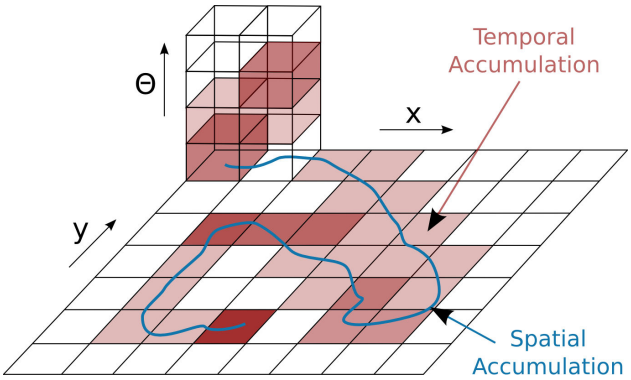
Approche heuristique Le travail de [Pro+10] utilise une technique heuristique d'intégration basée sur l'accumulation spatio-temporelle des réponses (voir figure III.14b) : le système classe chaque image indépendamment et ses réponses sont accumulées pendant un certain temps pour chaque pose (x, y, θ). De plus, chaque classification est pondérée par la confiance du système en celle-ci (en utilisant les techniques de [PC07]). La réponse du système est obtenue en sommant les valeurs pondérées. Cette technique a l'avantage d'être simple et de ne pas nécessiter d'apprentissage supplémentaire. Cependant, la taille des cases et la durée d'intégration doivent être choisies avec soin et le système met un certain temps avant de produire une réponse correcte. De plus, le problème de la détection des changements de lieu devient crucial : l'accumulation des mesures au cours du temps peut conduire à des erreurs lorsque le robot passe entre deux lieux. Pour pallier cela, un mécanisme de détection des portes, basé sur les capteurs lasers, permet de remettre à zéro l'histogramme des réponses. Enfin, le système dépend de manière critique des informations d'odométrie pour estimer la pose (x, y, θ) et nécessite l'acquisition d'une carte métrique.

Détection des changements de pièces Les travaux de RANGANATHAN [Ran10] utilisent une technique appelée *change-point detection*. L'idée générale est de détecter des changements dans la statistique du signal qui signalent qu'on est passé d'une pièce à l'autre. L'approche de [Ran10] se base sur un modèle : la forme de la distribution de probabilité du signal est supposée rester la même entre les différents lieux, seuls les paramètres changent.

Plus précisément, on suppose qu'une suite de mesures $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$ peut être découpée en segments adjacents et disjoints. La longueur du segment courant au temps t est notée par l_t. Le



(a) Étiquetage de lieux avec les techniques de [Moz+06].



(b) Technique d'intégration temporelle de [Pro+10].

FIGURE III.14.: (a) Technique d'intégration de [Moz+06] : la figure de gauche montre la classification de chaque cellule avant relaxation ; la figure de droite montre la classification après relaxation (les parties mal classées sont entourées en blanc) ; la classe jaune correspond aux transitions (« doorway »). (b) : Technique d'intégration de [Pro+10]. Extrait des références respectives.

but est de calculer la probabilité jointe $P(l_t, y_t|\boldsymbol{x}_{1:t})$. Dans l'approche de [Ran10], la probabilité de changement au temps t est prise constante donc l_t suit une distribution exponentielle. La probabilité de changement joue un rôle analogue à la probabilité de changement d'état dans le filtrage bayésien.

Le principal avantage de cette approche est que, comme on modélise la distribution jointe $P(l_t, y_t|\boldsymbol{x}_{1:t})$, le robot peut réviser l'ensemble de la segmentation et de la classification à tout moment. De plus, le modèle peut fonctionner sans données étiquetées (on peut découvrir de nouveaux lieux et l'apprentissage est incrémental).

Cependant, cela ne peut être obtenu que parce que le robot retient toutes les entrées sensorielles : la quantité de mémoire augmente linéairement en fonction du temps, ce qui constitue un problème majeur pour un robot devant opérer continuellement. Il existe cependant des algorithmes approchés nécessitant une mémoire constante. Néanmoins, un robot n'a sans doute pas besoin de pouvoir réviser à tout moment la classification de lieux. De plus, comme nous l'avons dit, les transitions entre seuils ne sont pas définies de manière parfaite. Il existe en fait une zone plus ou moins grande de transition entre deux lieux. Dès lors, chercher à segmenter complètement et uniquement la séquence d'images semble inutilement complexe. Cette technique, bien adaptée dans le cas des séquences de taille finie, semble relativement compliquée dans le cas de la reconnaissance de lieux et n'apporte pas d'amélioration significative des performances (voir Section III.4.1).

5. Conclusion

Ce chapitre s'est attaché à poser le problème de la reconnaissance de lieux sémantiques et à le distinguer du SLAM (dans lequel les lieux n'ont pas de caractère sémantique) et de la reconnaissance de scènes (dans lequel on reconnaît le contenu de l'image). Nous avons aussi expliqué en quoi le problème de la reconnaissance de lieux peut être utile pour un robot.

Nous avons ensuite présenté les différentes techniques développées dans le cadre de la reconnaissance de lieux. Il ressort de l'état de l'art que la tâche de reconnaissance d'instances a été bien étudiée, même si la robustesse des méthodes n'est pas toujours prouvée. En revanche, la tâche de catégorisation est toujours ouverte.

En ce qui concerne le meilleur codage, il est difficile de trancher car les méthodes sont rarement évaluées sur la même base. Cependant, les méthodes sémantiques (*i.e.* basées sur la reconnaissance préalable des objets) semblent trop compliquées pour le problème de reconnaissance. Les méthodes basées sur les mots visuels offrent de bonnes performances.

De manière générale, un effort important doit être fait pour évaluer les systèmes. Des domaines comme le traitement de la langue naturelle, la reconnaissance d'objets ou le SLAM ont pu bénéficier de bases de données partagées et de campagnes d'évaluation qui ont permis de comparer les travaux sur une base commune, faisant progresser la recherche. C'est pourquoi nous utiliserons les bases COLD et VPC qui sont les plus complètes à l'heure actuelle.

Les prochains chapitres seront consacrés à la présentation de notre travail.

Bibliographie

Introduction

[BKR07] E. Brunskill, T. Kollar et N. Roy. « Topological mapping using spectral clustering and classification ». Dans : *IROS*. 2007, p. 3491–3496 (cf. p. 120).

[BS02] P. Buschka et A. Saffiotti. « A virtual sensor for room detection ». Dans : *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. T. 1. IEEE. 2002, p. 637–642 (cf. p. 120).

[GB10] C. Galleguillos et S. Belongie. « Context based object categorization : A critical survey ». Dans : *Computer Vision and Image Understanding* 114.6 (2010), p. 712–722 (cf. p. 121).

[MSB05] O. M. Mozos, C. Stachniss et W. Burgard. « Supervised learning of places from range data using adaboost ». Dans : *IEEE International Conference on Robotics and Automation*. T. 2. 2005 (cf. p. 120).

[Tor+03a] A. Torralba et al. *Context-based vision system for place and object recognition*. Rap. tech. Cambridge, MA, 2003 (cf. p. 121).

[Tor+03b] A. Torralba et al. « Context-based vision system for place and object recognition ». Dans : *Proceedings of the Nineth IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*. T. 1. IEEE Computer Society, oct. 2003, p. 273–280. isbn : 0-7695-1950-4 (cf. p. 121, 131, 136, 138, 140–142).

[Tor03] A. Torralba. « Contextual priming for object detection ». Dans : *International Journal of Computer Vision* 53.2 (2003), p. 169–191 (cf. p. 121).

Position du problème

[KW95] D. Kortenkamp et T. Weymouth. « Topological mapping for mobile robots using a combination of sonar and vision sensing ». Dans : *Proceedings of the National Conference on Artificial Intelligence*. JOHN WILEY & SONS LTD. 1995, p. 979–979 (cf. p. 122).

[TS05] A. Tapus et R. Siegwart. « Incremental robot mapping with fingerprints of places ». Dans : *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE. 2005, p. 2429–2434 (cf. p. 122).

[Ull+07] M. M. Ullah et al. *The COLD Database*. Rap. tech. TRITA-CSC-CV 2007 :1. Kungliga Tekniska Högskolan, CVAP/CAS, oct. 2007 (cf. p. 125, 127–131).

[UN00] I. Ulrich et I. Nourbakhsh. « Appearance-Based Place Recognition for Topological Localization ». Dans : *IEEE International Conference on Robotics and Automation*. San Francisco, CA, 2000, p. 1023–1029 (cf. p. 122).

[VLD06] C. Valgren, A. J. Lilienthal et T. Duckett. « Incremental Topological Mapping Using Omnidirectional Vision ». Dans : *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2006, p. 3441–3447 (cf. p. 122).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 124, 137, 140).

Challenges

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 129, 132).

[Tor+03b] A. Torralba et al. « Context-based vision system for place and object recognition ». Dans : *Proceedings of the Nineth IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*. T. 1. IEEE Computer Society, oct. 2003, p. 273–280. isbn : 0-7695-1950-4 (cf. p. 121, 131, 136, 138, 140–142).

[Ull+07] M. M. Ullah et al. *The COLD Database*. Rap. tech. TRITA-CSC-CV 2007 :1. Kungliga Tekniska Högskolan, CVAP/CAS, oct. 2007 (cf. p. 125, 127–131).

État de l'art en reconnaissance de lieux sémantiques

[CN08] M. Cummins et P. Newman. « FAB-MAP : Probabilistic localization and mapping in the space of appearance ». Dans : *The International Journal of Robotics Research* 27 (2008), p. 647–665 (cf. p. 134, 138).

[CN11] M. Cummins et P. Newman. « Appearance-only SLAM at large scale with FAB-MAP 2.0 ». Dans : *The International Journal of Robotics Research* 30 (2011), p. 1100–1123 (cf. p. 134).

[FEN07] F. Fraundorfer, C. Engels et D. Nistér. « Topological mapping, localization and navigation using image collections ». Dans : *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, San Diego, California, USA*. IEEE. 2007, p. 3872–3877 (cf. p. 134, 141).

[Fil08] D. Filliat. « Interactive learning of visual topological navigation ». Dans : *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*. 2008 (cf. p. 138, 140, 141).

[FPF07] S. Friedman, H. Pasula et D. Fox. « Voronoi random fields : Extracting the topological structure of indoor environments via place labeling ». Dans : *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2007 (cf. p. 143).

[GO00] A. Guérin-Dugué et A. Oliva. « Classification of scene photographs from local orientation features ». Dans : *Pattern Recognition Letters* 21 (2000), p. 1135–1140 (cf. p. 138).

[KF04] J. Kosecka et L. Fayin. « Vision Based Topological Markov Localization ». Dans : *IEEE International Conference on Robotics and Automation*. New Orleans, LA, USA, 2004 (cf. p. 138, 142).

[Moz+06] O. M. Mozos et al. « Semantic Labeling of Places using Information Extracted from Laser and Vision Sensor Data ». Dans : *Proceedings of the IEEE/RSJ IROS Workshop : From sensors to human spatial concepts*. Beijing, China, 2006, p. 2772–2777. ISBN : 1-4144-0259-X (cf. p. 138, 143, 144).

[Moz+07] O. M. Mozos et al. « Supervised semantic labeling of places using information extracted from sensor data ». Dans : *Robotics and Autonomous Systems* 55.5 (2007), p. 391–402 (cf. p. 141, 142).

[MW08] M. Milford et G. Wyeth. « Mapping a suburb with a single camera using a biologically inspired SLAM system ». Dans : *IEEE Transactions on Robotics* 24 (2008), p. 1038–1053 (cf. p. 134).

[Ora+07] F. Orabona et al. « Indoor Place Recognition using Online Independent Support Vector Machines ». Dans : *Proceeding of the British Machine Vision Conference (BMVC 2007)*. Warwick, UK, 2007, p. 1090–1099 (cf. p. 134, 138).

[PC07] A. Pronobis et B. Caputo. « Confidence-based cue integration for visual place recognition ». Dans : *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, San Diego, California, USA*. Oct. 2007 (cf. p. 134, 138–141, 143).

[Pro+10] A. Pronobis et al. « Multi-modal semantic place classification ». Dans : *The International Journal of Robotics Research* 29.2-3 (2010), p. 298–320 (cf. p. 136, 143, 144).

[QT09] A. Quattoni et A. Torralba. « Recognizing Indoor Scenes ». Dans : *IEEE Conference on Computer Vision and Pattern Recognition*. 2009 (cf. p. 138).

[Ran10] A. Ranganathan. « PLISS : Detecting and Labeling Places using Online Change-point Detection ». Dans : *Proceedings of the 2010 Robotics : Science and Systems Conference (RSS 2010)*. 2010 (cf. p. 136–138, 140, 143, 145).

[RD07] A. Ranganathan et F. Dellaert. « Semantic modeling of places using objects ». Dans : *Robotics : Science and Systems*. Citeseer. 2007 (cf. p. 138, 139, 142).

[Tor+03b] A. Torralba et al. « Context-based vision system for place and object recognition ». Dans : *Proceedings of the Nineth IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*. T. 1. IEEE Computer Society, oct. 2003, p. 273–280. ISBN : 0-7695-1950-4 (cf. p. 121, 131, 136, 138, 140–142).

[Ull+08] M. M. Ullah et al. « Towards Robust Place Recognition for Robot Localization ». Dans : *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008)*. Pasadena, USA, mai 2008 (cf. p. 134–137, 139, 140).

[Vis+10] P. Viswanathan et al. « Automated Place Classification using Object Detection ». Dans : *2010 Canadian Conference Computer and Robot Vision*. IEEE. 2010, p. 324–330 (cf. p. 138, 139).

[VL10] C. Valgren et A. Lilienthal. « SIFT, SURF & seasons : Appearance-based long-term localization in outdoor environments ». Dans : *Robotics and Autonomous Systems* 58.2 (2010), p. 149–156 (cf. p. 134, 138).

[VS07] S. Vasudevan et R. Siegwart. « A Bayesian approach to Conceptualization and Place Classification : Incorporating Spatial Relationships (distances) to Infer Concepts ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07) workshop "From Sensors to Human Spatial Concepts (FS2HSC)"*. San Diego, USA, 2007 (cf. p. 138–140).

[WBB02] J. Wolf, W. Burgard et H. Burkhardt. « Using an Image Retrieval System for Vision-Based Mobile Robot Localization ». Dans : *Proceedings of the 1st ACM International Conference on Image and Video Retrieval (CIVR 2002)*. Sous la dir. de M. S. Lew, N. Sebe et J. P. Eakins. T. 2383. Lecture Notes in Computer Science. London, UK : Springer, juil. 2002. ISBN : 3-540-43899-8 (cf. p. 134, 141).

[WCR09] J. Wu, H. Christensen et J. Rehg. « Visual place categorization : Problem, dataset, and algorithm ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE. St. Louis, USA, 2009, p. 4763–4770 (cf. p. 136, 138, 142).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 124, 137, 140).

IV. tBoW : Reconnaissance de lieux avec des signatures globales et des modèles sac de mots

Nous présentons dans ce chapitre un premier modèle de reconnaissance de lieux, développé à la suite des travaux de thèse de Hervé Guillaume (voir [Gui09]). Nous présentons et évaluons en détail le modèle. Les chapitres suivants seront consacrés à des extensions de ces travaux.

1. Introduction

L'état de l'art développé dans la section III.4 fait apparaître les points suivants :

1. La reconnaissance de lieux ne nécessite pas de décomposer une scène ou un lieu en objets : cette étape est d'un coût prohibitif et n'a d'utilité que dans le cadre d'un raisonnement de haut niveau ou dans le cadre d'une interaction motrice complexe avec l'environnement. Au contraire, l'identification préalable du contexte sémantique peut permettre de guider le robot et éventuellement faciliter le processus de reconnaissance des objets. Notre travail n'utilisera donc pas de méthodes basées sur la reconnaissance préalable des objets.
2. Au niveau de la caractérisation des images, l'utilisation de descripteurs globaux est suffisante. Par ailleurs, les méthodes Bag-of-Words (BoW) permettent de simplifier grandement l'apprentissage tout en fournissant des systèmes au niveau de l'état de l'art. Notre travail utilisera donc les méthodes basées sur les mots visuels, décrites dans la section II.3.
3. Les approches bayésiennes de l'intégration temporelle fournissent un cadre probabiliste cohérent pour le problème : nous présenterons plusieurs méthodes inspirées de ce cadre. De plus, ces méthodes sont compatibles avec l'approche générative en apprentissage. Nous avons vu à la Section I.3.2 que plusieurs arguments plaident en faveur des modèles d'apprentissage génératifs dans le cadre de la robotique autonome. Citons, entre autres, la nécessité de disposer d'une technique d'apprentissage incrémentale, qui ne nécessite pas la présentation de contre-exemples pour généraliser l'extension d'une classe dans l'espace de représentation des entrées.

À partir de ces constats, Guillaume [Gui09] a proposé un système de reconnaissance de lieux qui forme la base de notre travail. Cependant, l'évaluation du système a été faite sur une base de données ne contenant que deux séquences (une séquence d'entraînement et une séquence de reconnaissance), sans prise en compte des variations de luminosité et ne contenant que deux catégories (« bureau » et « couloir »). Les résultats obtenus doivent donc être confirmés sur des bases plus générales. Nous chercherons donc à tester les propriétés de notre système et à les mettre en relation avec un certain nombre de travaux réalisés indépendamment.

Nous présentons tout d'abord les expériences préliminaires qui ont permis de concevoir le système (section IV.2) puis nous présentons l'architecture de celui-ci (section IV.3). Les parties IV.4 et IV.5 sont consacrées à l'étude des paramètres du système et son évaluation sur les tâches de reconnaissance d'instances et de catégorisation respectivement.

2. Codages globaux et catégories sémantiques

Dans un système d'apprentissage supervisé, la représentation de l'environnement est généralement fixée *a priori*. La pertinence de ce codage est validée par les résultats en classification. Dans le cas de la reconnaissance de lieux sémantiques, les caractéristiques pertinentes ne sont pas évidentes.

Si un codage est pertinent, les liens entre différents lieux doivent apparaître spontanément. Un algorithme de classification non-supervisée peut mettre à jour ces liens.

Les travaux de Guillaume, Denquive et Tarroux [GDT05], étendus dans Guillaume [Gui09], ont étudié la pertinence de codages globaux pour la reconnaissance de lieux. Le but était de savoir si les prototypes qui émergent lors de la phase de classification non-supervisée permettent de distinguer différentes catégories d'environnements. Les environnements intérieurs sont visuellement plus homogènes donc la discrimination inter-catégories est plus difficile.

Les images sont issues de l'exploration du LIMSI par un robot. Les codages utilisés étaient de deux types :

1. Codage basé sur les caractéristiques fréquentielles : cette signature est basée sur l'énergie moyenne en sortie d'un banc de filtres de Gabor composé de 4 orientations $\{0, \pi/4, \pi/2, 3\pi/4\}$ et 5 fréquences spatiales $\{1/5, 1/10, 1/20, 1/40, 1/80\}$ $^{\text{cycles}}/_{\text{pixel}}$. Ce codage est similaire au descripteur GIST.
2. Codage à l'aide de fonctions de texture : ces signatures sont basées sur un sous-ensemble des critères de Haralick (voir [HSD73]) que nous avons déjà évoqué dans la Section I.2.1.3. Notons que, contrairement à ce que nous faisons ici, ces grandeurs sont normalement calculées dans un voisinage autour de chaque pixel (elles caractérisent la texture en chaque point de l'image). Différentes alternatives ont été explorées : calcul sur le canal de luminance seul, sur les canaux de couleurs sans information de luminance ou utilisation du canal de luminance et des canaux de couleurs simultanément.

Les prototypes (qui correspondent aux mots visuels des méthodes BoW) sont obtenus de manière non-supervisée au moyen d'une carte auto-organisatrice, appelée *Self-Organizing Map (SOM)*. Une explication détaillée de l'algorithme d'apprentissage est donnée en annexe (voir Appendice B).

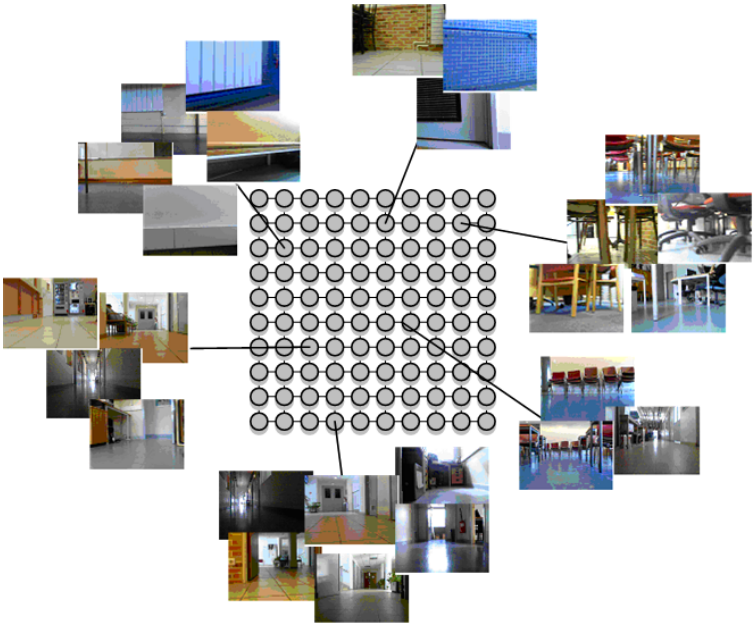
Les SOM (voir [Koh89]) modélisent les données d'entrée à l'aide de prototypes, ce qui permet de partitionner l'espace d'entrée, $\mathcal{X}$, en groupes de données similaires. De ce point de vue, c'est un algorithme classique de clustering comme peut l'être, par exemple, l'algorithme des K-moyennes.

Cependant, la propriété la plus intéressante des cartes de Kohonen par rapport à d'autres algorithmes de quantification vectorielle, tels que les K-moyennes, est de préserver les relations topologiques entre les vecteurs de l'espace d'entrée : les prototypes (ou nœuds) proches sur la carte sont également proches dans l'espace d'entrée. Cette propriété permet de représenter des données de dimension N dans un espace discret de dimension 2. Il est alors plus simple de visualiser et d'interpréter la « logique » sous-jacente au partitionnement des données, ce qui facilite l'interprétation des résultats. De plus, il est possible de définir des groupements d'unités qui forment des clusters de plus haut niveau (voir [VA02]).

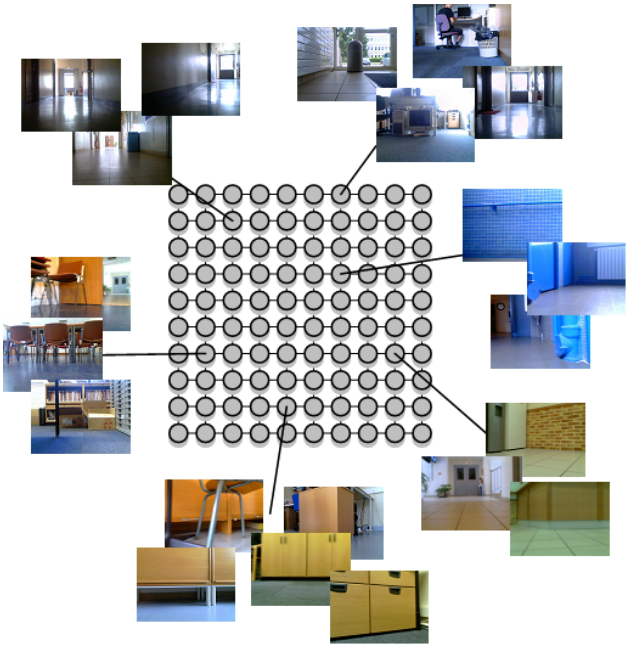
En ce qui concerne les environnements d'intérieur structurés, les prototypes obtenus par les deux types de codage ne permettent pas de discriminer les différentes catégories de lieux. On observe en effet (voir Figure IV.1) que les images de différents lieux sont réparties de manière quasi-uniforme sur la carte. En revanche, les prototypes regroupent des images caractéristiques visuellement proches, qui dépendant à la fois du point de vue de l'observateur et de la scène observée : certaines cellules regroupent les images correspondantes à des scènes ouvertes telles l'entrée d'une pièce, d'autres à des vues plus fermées (par exemple lorsque le robot est proche des murs, *etc.*).

Ces prototypes ressemblent aux catégories intermédiaires étudiées par Oliva et Torralba [OT01]. On les appelle parfois « catégories syntaxiques » par opposition aux catégories sémantiques que forment les lieux. Il a de plus été montré que la catégorie syntaxique d'une image véhicule des informations sur les objets que l'on peut trouver dans l'image (voir [Gui09, chap. 5]).

En résumé, les cartes de Kohonen permettent de créer un vocabulaire visuel à partir de signatures globales pour les environnements intérieurs. Les prototypes obtenus ne sont pas caractéristiques d'un lieu ou d'une catégorie de lieux mais forment des classes de scènes visuelles. Ces travaux ont conduit Guillaume [Gui09] à proposer une architecture de reconnaissance de lieux que nous présentons dans la section suivante.



(a) Codage basé sur les caractéristiques fréquentielles.



(b) Codage basé sur les fonctions de texture.

FIGURE IV.1.: Clustering effectué par une carte de 10 unités de côtés sur des environnements d'intérieur structurés. La figure montre les images types associées à certaines unités : (a) pour le codage basé sur les caractéristiques fréquentielles (b) pour le codage basé sur les fonctions de texture. On observe que des images similaires en provenance de différents lieux sont associées à la même cellule. Extrait de [Gui09].

3. Architecture du système

Comme nous l'avons dit, une vue (c'est-à-dire une image) peut être intrinsèquement ambiguë, non-informative ou même trompeuse pour la reconnaissance de lieux. En tout état de cause, une seule vue est rarement suffisante pour reconnaître un lieu. En revanche, un lieu particulier, défini sémantiquement, peut être décrit par un ensemble de vues collectées au cours du temps. Cette observation est à la base du système que nous proposons.

3.1. Vue d'ensemble

La Figure IV.2 présente une vue d'ensemble du système. Chaque image extraite du flux vidéo est caractérisée à l'aide d'une signature globale.

L'apprentissage a lieu en deux étapes. Premièrement, les signatures sont utilisées pour entraîner une carte de Kohonen, qui nous fournit un ensemble de K prototypes. Comme l'ont montré les expériences décrites dans la section IV.2, ces prototypes seuls ne permettent pas de reconnaître les lieux. Quand une image est présentée, sa signature est calculée puis attribuée au prototype le plus proche. Chaque image est donc représentée par un entier $k \in \{1, \ldots, K\}$.

Ces prototypes sont utilisés dans une phase d'apprentissage pour créer des modèles de lieux. Nous utilisons des modèles génératifs permettant d'estimer la vraisemblance de chaque prototype, donc de chaque image pour un lieu donné. Finalement, les différents indices récoltés au cours du temps sont intégrés afin de donner une réponse.

Un lieu sémantique, est décrit comme un ensemble de mots visuels globaux, chacun décrivant une scène, de la même manière qu'une phrase est construite à partir de mots. Dans un premier temps, nous ne tenons pas compte de l'ordre des mots : notre modèle rentre donc dans la famille des modèles BoW.

Comme nous l'avons dit (voir section II.3), ces méthodes consistent à décrire une image à l'aide d'une collection de mots. L'utilisation d'une suite d'images nous permet de combiner une caractérisation globale des images et les méthodes BoW : au lieu de collecter les mots grâce à une caractérisation locale de l'image, les mots sont collectés temporellement. C'est pourquoi nous avons baptisé notre système *temporal Bag-of-Words (tBoW)*. Cette idée, profondément reliée au problème de la reconnaissance de lieux, permet de diminuer l'ambiguïté associée à une image en s'appuyant sur la continuité temporelle (voir Section III.3.3). On tire ainsi avantage des capacités motrices et du caractère situé du robot.

Dans les méthodes BoW, une image est décrite par la fréquence d'apparition des différents mots visuels qui s'y trouvent. Le modèle d'une classe et la reconnaissance utilisent ce vecteur de représentation. De même, les modèles de lieux utilisés dans tBoW se basent sur la fréquence d'apparition d'un prototype dans chaque lieu. La reconnaissance fait appel à un ensemble d'images.

Afin d'étudier quels traits sont pertinents pour la reconnaissance de lieux, nous utilisons

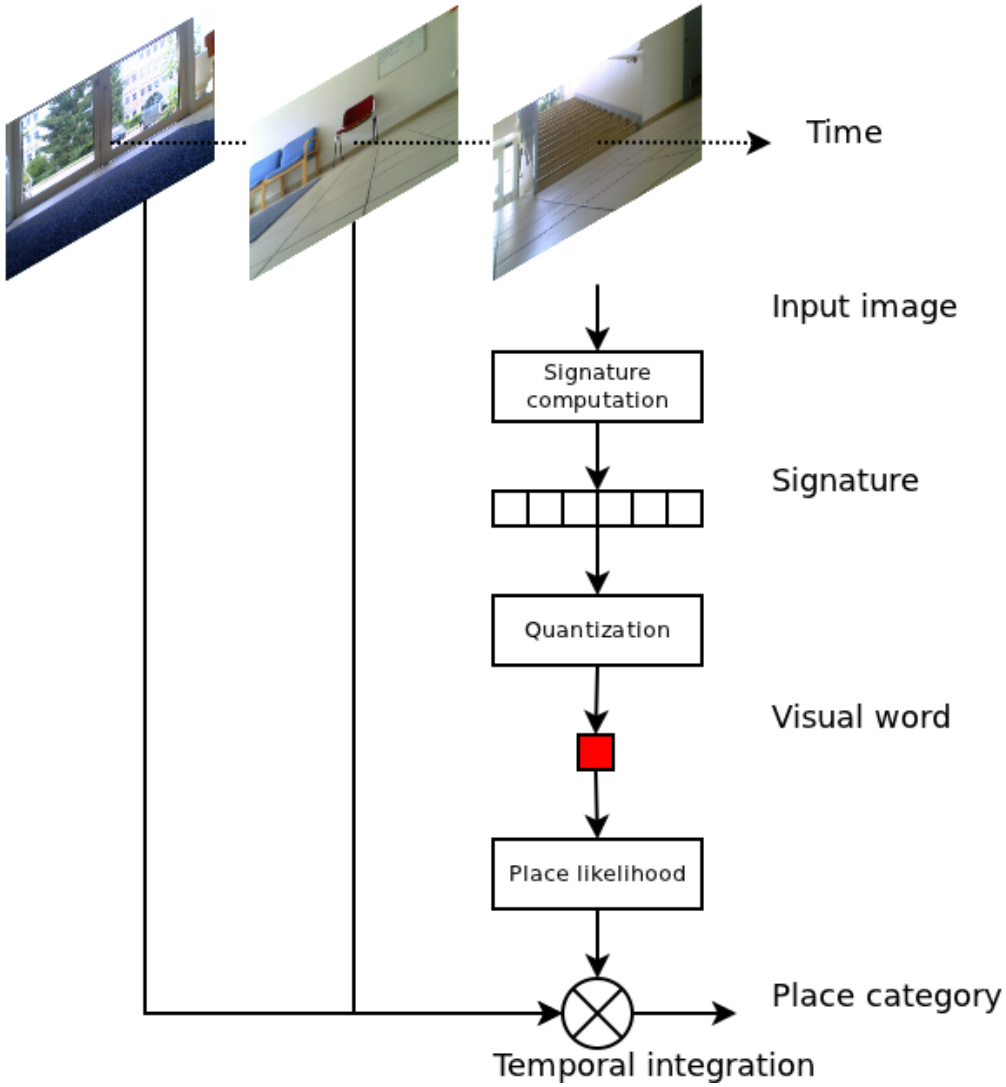


FIGURE IV.2.: Architecture du système de reconnaissance de lieux proposée par [Gui09]. Les entrées visuelles sont caractérisées globalement au cours de l'exploration. Une étape de quantification vectorielle permet de manipuler un nombre fini de prototypes visuels. Un lieu ou une catégorie de lieu est décrite en fonction de la fréquence relative d'apparition des prototypes, profitant du caractère situé et des capacités exploratoires du robot pour reconnaître un lieu au travers d'une séquence temporelle plutôt qu'à partir d'une image fixe.

plusieurs types de signatures que nous décrivons dans la section suivante. L'entraînement de la carte de Kohonen est décrit dans la Section IV.3.3. La manière d'utiliser plusieurs images est présentée dans la Section IV.3.4.

3.2. Signatures utilisées

Nous utilisons plusieurs types de signatures :
- les signatures basées sur les fonctions de texture qui sont les mêmes que Guillaume [Gui09]
- le descripteur GIST proposé par Torralba et al. [Tor+03]
- le descripteur CENTRIST proposé par Wu, Christensen et Rehg [WCR09]

Cela nous permettra d'une part de savoir quels traits sont les plus pertinents pour les différentes tâches et d'autre part de comparer notre système à d'autres travaux de la littérature.

3.2.1. Signatures basées sur les fonctions de texture

Ces signatures ont été décrites en détail dans la section IV.2 et dans l'Appendice C.

Les résultats de Guillaume [Gui09] ont montré que les meilleurs résultats ont été obtenus en utilisant seulement le canal de luminance. Cependant, on peut postuler que cette signature n'est pas robuste au changement de condition d'éclairement.

3.2.2. GIST

GIST est décrit en détail dans la Section I.2.2.1. Comme nous l'avons dit, on peut calculer le descripteur sur le canal de luminance seul ou sur les canaux de couleur. Nous travail utilise les deux possibilités.

Par ailleurs, nous utilisons $n_\sigma = 4$ échelles et $n_\theta = 6$ orientations par échelle. Pour calculer l'énergie du banc de filtre, nous décomposons chaque image en 16 sous-images (*i.e.* $M = 4$).

Le descripteur GIST est habituellement projeté sur un nombre plus faible de dimensions en utilisant une ACP. Torralba et al. [Tor+03] utilisent 80 dimensions, sans justifier cette valeur. Or, il existe plusieurs approches pour déterminer le nombre de dimensions à utiliser lors de la projection :
- la règle de Kaiser qui veut qu'on ne retienne que les dimensions dont les valeurs propres sont supérieures à 1. Nous n'avons pas retenu la méthode de Kaiser dans nos travaux car elle a tendance à sous-estimer le nombre de dimensions utiles.
- le test du coude (*scree-test*) qui se base sur une analyse de la courbe des valeurs propres : cette courbe présente en général un point d'inflexion (un « coude ») ; la règle consiste à retenir les valeurs situées à gauche du point d'inflexion. Cette méthode est très utile quand le nombre de dimensions initial est faible (de l'ordre de la dizaine). Dans notre cas, elle s'est avérée difficile à estimer correctement à cause de la dimensionnalité du

	Nombre de dimensions		
Canaux utilisés	**Avant projection**	**95% variance**	[Tor+03]
Luminance	384	22	80
Couleurs	1152	17	

TABLE IV.1.: Nombre de dimensions du descripteur GIST avant et après projection sur une base issue de l'ACP.

descripteur. Par exemple le descripteur sur le canal de luminance possède 384 dimensions. Même si les valeurs propres associées décroissent rapidement, il existe de nombreuses valeurs propres très faibles qui rendent difficile la détection précise du coude. Nous avons donc choisi de ne pas l'utiliser.

– choisir le nombre d'axes en fonction de la restitution minimale d'information que l'on souhaite retenir : on peut par exemple retenir les dimensions qui expliquent 95% de la variance. Ce critère est très simple à évaluer et la valeur est fondée statistiquement. Le tableau IV.1 donne le nombre de dimensions nécessaires pour expliquer 95% de la variance. On voit que ce nombre est nettement inférieur à la valeur fixée par TORRALBA et al.

Il nous a semblé intéressant de comparer les résultats obtenus en projetant les signatures sur le nombre de dimensions retenu par la règle des 95 % de variance et ceux obtenus en utilisant 80 dimensions (comme [Tor+03]). La base de projection a été obtenue en utilisant un échantillon constitué de 1/10 des images des bases COLD et VPC (soit plus de 18 000 images).

3.2.3. CENTRIST

Nous utilisons aussi CENTRIST décrit dans la Section I.2.2.2. Contrairement à WU, CHRISTENSEN et REHG [WCR09], nous calculons l'histogramme de la CT sur toute l'image. On obtient ainsi un descripteur de 256 dimensions.

Nous avons considéré l'utilisation d'une ACP pour diminuer le nombre de dimensions du descripteur. Cependant, les tests statistiques utilisés pour vérifier la validité de l'ACP (test de Bartlett et test KMO [Kai74]) ont été négatifs (la valeur de l'indice KMO étant de l'ordre de 0,22, ce qui est très faible). Nous avons donc choisi de ne pas réduire la dimensionnalité de ce descripteur.

3.3. Entraînement des cartes de Kohonen

Une fois les signatures calculées, nous devons en utiliser un échantillon pour entraîner la carte de Kohonen. Dans ce chapitre et dans les suivants, cet entraînement est réalisé avant l'apprentissage des modèles de lieux proprement dit. Les paramètres et les jeux de données utilisés sont décrit dans la section B.3. Soulignons que nous utilisons des images issues de toutes les conditions, notamment de toutes les conditions de luminance. On obtient ainsi un dictionnaire « universel » car il est représentatif de l'ensemble des images et « inné » car son entraînement n'est effectué qu'une seule fois, avant l'apprentissage des lieux proprement dit.

Les cartes de Kohonen sont un élément-clé du système car elles fournissent les mots visuels qui vont nous permettre d'apprendre les modèles de lieux. Parmi les paramètres, la taille de la SOM, S, définit le nombre de clusters : $K = S \times S$. C'est un paramètre important du système dont nous étudierons l'effet sur les performances.

3.4. Apprentissage et intégration temporelle

3.4.1. Principe de la méthode

Comme nous l'avons dit, le système est inspiré des techniques de type BoW. Une fois l'image courante transformée en un mot visuel, il est nécessaire d'apprendre un modèle des lieux basé sur ces prototypes. La règle d'apprentissage et d'intégration que nous proposons est une simplification du filtrage bayésien. Le chapitre VI proposera une autre méthode.

Comme nous l'avons dit (voir section III.2), d'un point de vue probabiliste, le but est de prédire la probabilité d'être en un lieu connaissant l'historique des observations $\boldsymbol{x}_{1:t}$: $\mathrm{P}\left(y_t|\boldsymbol{x}_{1:t}\right)$.

Dans notre cas, l'observation est un scalaire x_t, qui correspond à une cellule k de l'ensemble discret et fini $\{1, \ldots, K\}$ des cellules de la carte de Kohonen. Le système ne fait pas appel à tout l'historique mais seulement à une fenêtre de taille fixe $x_{t-L+1:t}$ constituée des L dernières observations $\{x_{t-L+1}, \ldots, x_{t-1}, x_t\}$. On fait donc l'hypothèse que $\mathrm{P}\left(y_t|x_{1:t}\right) = \mathrm{P}\left(y_t|x_{t-L+1:t}\right)$. Pour les instants $t < L$, la fenêtre d'intégration est tronquée à $\max\{t - L + 1, 1\}$.

La formule de Bayes nous donne :

$$\mathrm{P}\left(y_t|x_{t-L+1:t}\right) = \frac{\mathrm{P}\left(x_{t-L+1:t}|y_t\right)\mathrm{P}\left(y_t\right)}{\mathrm{P}\left(x_{t-L+1:t}\right)} \tag{IV.1}$$

On fait de plus l'hypothèse que, connaissant y_t, les L dernières observations sont indépendantes. On obtient donc la règle d'intégration temporelle suivante :

$$\mathrm{P}\left(x_{t-L+1:t}|y_t\right) = \prod_{t'=t-L+1}^{t} \mathrm{P}\left(x_{t'}|y_t\right) \tag{IV.2}$$

L'hypothèse d'indépendance implique que, dans un lieu donné, l'ordre d'apparition des mots ne compte pas : seules leurs vraisemblances dans le lieu courant $\mathrm{P}\left(x_{t'}|y_t\right)$ comptent. Elle est analogue à l'hypothèse du BoW qui stipule qu'on peut décrire une image sans prendre en compte les relations géométriques entre les mots. Cette hypothèse d'indépendance des observations peut paraître très forte. Cependant si l'on considère que le robot peut parcourir une pièce dans différents sens et éventuellement commencer son exploration à partir de n'importe quel point de la pièce, prendre en compte l'ordre des mots pourrait, au contraire, nuire à la qualité de la reconnaissance.

D'un point de vue de l'apprentissage, l'hypothèse d'indépendance des observations est similaire à l'hypothèse du Classificateur Bayésien Naïf (voir Section I.3.2).

La différence par rapport au filtrage bayésien vient du fait qu'il n'y a pas de modélisation des changements d'état (c'est-à-dire de lieu). On suppose que le lieu courant génère les L dernières observations. Le système détecte lui-même l'état courant sur la base de ces observations : le lieu le plus probable est celui pour lequel la probabilité *a posteriori* est la plus forte. Rappelons que dans les approches bayésiennes, la probabilité de changement d'états est liée à la topologie de l'environnement (voir Section III.4.4.1). En ignorant la modélisation des changements de lieu, la reconnaissance ne dépend pas de la topologie de l'environnement d'apprentissage.

Classification Pour la classification, on cherche le lieu $\overset{\star}{y}_t$ qui maximise la probabilité *a posteriori*. En combinant l'Équation IV.1 et l'Équation IV.2, on a :

$$\begin{aligned}\overset{\star}{y}_t &= \underset{c^i \in \mathcal{C}}{\arg\max} \left\{ \mathrm{P}\left(y_t = c^i | x_{t-L+1:t}\right) \right\} \\ &= \underset{c^i \in \mathcal{C}}{\arg\max} \left\{ \mathrm{P}\left(y_t = c^i\right) \prod_{t'=t-L+1}^{t} \mathrm{P}\left(x_{t'}|y_t\right) \right\} \end{aligned} \tag{IV.3}$$

En considérant que les lieux sont équiprobables, on peut appliquer la règle du maximum de vraisemblance (voir Équation I.13, page 54). On obtient donc la règle de classification du système :

$$\overset{\star}{y}_t = \underset{c^i \in \mathcal{C}}{\arg\max} \left\{ \prod_{t'=t-L+1}^{t} \mathrm{P}\left(x_{t'}|y_t = c^i\right) \right\} \tag{IV.4}$$

Au-delà de la classification, il est intéressant de calculer la distribution de probabilité *a posteriori* $\mathrm{P}\left(y|x_{t-L+1:t}\right)$. Cette distribution permet de classer les lieux par ordre de probabilité. Cette propriété sera utilisée dans le chapitre V. Nous détaillerons le calcul dans la Section IV.3.4.3.

Apprentissage des paramètres La vraisemblance $\mathrm{P}(x_{t'}|y)$ est obtenue par apprentissage. Comme nous utilisons un apprentissage génératif, nous apprenons directement cette distribution de probabilité. Ici, l'utilisation de la phase préalable de quantification montre clairement ses avantages. En effet, l'observation x_t est discrète alors que la signature elle-même est un vecteur dans un espace de grande dimension. L'estimation de la probabilité est donc beaucoup plus facile grâce à la discrétisation opérée par la carte de Kohonen.

Le modèle du lieu c^i est décrit par un vecteur $\boldsymbol{P^i}$, tel que $\boldsymbol{P^i}[k] = \mathrm{P}(x_t = k|y_t = c^i)$. L'estimation de la distribution se fait grâce à la formule de Lidstone (connue aussi sous le nom de loi de Laplace, voir [CM10, chap. 15]) :

$$\boldsymbol{P^i}[k] = \frac{m_k + M}{m + M \times K} \tag{IV.5}$$

où :

- m^i est la taille de l'échantillon d'apprentissage du lieu c^i.
- m^i_k est le nombre de fois où le prototype k a été observé dans le lieu c^i.
- M est un entier positif arbitraire qui représente une augmentation virtuelle du nombre d'observations ; le terme $M \times K$ assure la normalisation. Cette étape de régularisation (*smoothing*) permet d'éviter d'avoir des probabilités nulles pour les mots non-observés dans l'échantillon d'apprentissage. En général, nous utilisons $M = 1$.

La probabilité *a priori* est décrite par un vecteur $\boldsymbol{\pi}$ tel que $\boldsymbol{\pi}[i] = \mathrm{P}(y_t = c^i)$. Nous supposons que cette probabilité est uniforme :

$$\boldsymbol{\pi}[i] = \frac{1}{p} \tag{IV.6}$$

Parmi les avantages de cette approche citons :

1. Elle est très simple à mettre en œuvre et extrêmement rapide.
2. L'apprentissage, bien que supervisé, est incrémental. Par conséquent, le robot peut revenir plusieurs fois dans un même lieu au cours de son exploration pour affiner progressivement les statistiques associées. Il peut également apprendre de nouveaux modèles de lieu en laissant les autres inchangés. Enfin, l'apprentissage peut être effectué en ligne, par exemple sous la conduite d'un « tuteur » indiquant au robot le lieu où il se trouve.
3. Il est facile de construire un modèle de catégorie (par exemple « bureau ») à partir des modèles d'instances de cette catégorie (par exemple « bureau 205 » et « bureau 206 ») : il suffit de sommer les termes m_k dans l'Équation IV.5 pour toutes les instances de la catégorie considérée. Par conséquent, la catégorisation ne nécessite pas d'apprentissage particulier : le modèle de la catégorie peut être obtenu directement à partir des modèles individuels.

Notons que si $L = 1$, le système se comporte comme un système de reconnaissance de scènes classique. On peut aussi envisager le cas particulier d'une longueur d'intégration "infinie", c'est-à-dire constituée par toutes les observations. Enfin, notons que le paramètre L ne joue aucun rôle dans l'apprentissage du modèle de lieu.

3.4.2. Problème de la détection des transitions

Nous avons insisté dans la présentation de l'état de l'art sur l'importance de la détection des transitions (voir section III.4). Or, la règle d'intégration que nous avons présentée ne tient pas compte des changements de lieu. Ainsi, lorsque le robot change de lieu, la fenêtre $x_{t-L+1:t}$ contient des prototypes issus du lieu précédent, ce qui risque de nuire à la reconnaissance.

D'autres auteurs ont été confrontés au problème et notre système pourrait donc utiliser les mêmes techniques (voir page 143) : détection des portes à l'aide de capteurs lasers ([Pro+10]), détection des points de changement ([Ran10]), *etc.*

Il est donc raisonnable, dans un premier temps, de traiter le problème de la détection des transitions à part. Afin de nous concentrer sur la caractérisation de notre système, nous supposons que le robot est informé, en phase de reconnaissance, quand il change de lieu. Lors de ce changement, les prototypes précédents sont oubliés (nous appellerons parfois ce mécanisme le « reset ») : la taille de la fenêtre est dynamique et L constitue une borne supérieure. Les travaux de Radhakrishnan et Nourbakhsh [RN99], montrent qu'il est possible d'obtenir de très bons systèmes de détection des transitions sur la base visuelle. Nous présenterons nos travaux sur ce problème dans le chapitre V.

Les travaux de [Gui09] ont montré que si l'on ne tient pas compte des changements de lieu, les performances sont nettement moins bonnes, au moins sur la tache de reconnaissance d'instances.

3.4.3. Implémentation

Si elle est implémentée naïvement, l'Équation IV.4 peut conduire à des problèmes numériques. En effet, il arrive fréquemment que les vraisemblances $\mathrm{P}\left(x_{t'}|y\right)$ soient faibles (de l'ordre de 10^{-3}). Or, nous utilisons fréquemment de nombreux termes (de l'ordre de la centaine). Dans ce cas, il peut arriver que les résultats du calcul des vraisemblances soient trop petits pour être représentés par la machine : on parle de soupassement arithmétique [1] (voir [Gol91]). En général, les résultats sont alors tronqués à 0 (au mieux, le programme est interrompu par une exception). Ce problème peut intervenir à n'importe quel instant t' dans l'Équation IV.4. Dès lors, quelle que soit la vraisemblance d'une observation ultérieure, la vraisemblance de la fenêtre est nulle. Dans le pire des cas, il se peut que toutes les vraisemblances soient nulles.

1. Notre lecteur toujours passionné d'étymologie, notera que le terme « soupassement » est un néologisme formé sur le modèle de « dépassement » pour traduire le néologisme anglais *underflow*. Source : http://fr.wikipedia.org/wiki/Soupassement_arithm%C3%A9tique.

Pour pallier cela, nous utilisons une formulation logarithmique du problème. Si l'on considère le logarithme de l'Équation IV.1, on a :

$$\log\left(\mathrm{P}\left(y_t|x_{t-L+1:t}\right)\right) = \log\left(\mathrm{P}\left(x_{t-L+1:t}|y_t\right)\right) + \log\left(\mathrm{P}\left(y_t\right)\right) - \log\left(\mathrm{P}\left(x_{t-L+1:t}\right)\right)$$

Le logarithme de la vraisemblance est obtenu par :

$$\log\left(\mathrm{P}\left(x_{t-L+1:t}|y_t\right)\right) = \sum_{t'=t-L+1}^{t} \log\left(\mathrm{P}\left(x_{t'}|y_t\right)\right) \tag{IV.7}$$

La probabilité *a priori* $\log\left(\mathrm{P}\left(y_t\right)\right)$ est supposée uniforme :

$$\log\left(\mathrm{P}\left(y_t = c^i\right)\right) = -\log\left(p\right), \forall c^i \in \left\{c^1, ..., c^p\right\} \tag{IV.8}$$

La probabilité marginale est obtenue classiquement à partir de la formule des probabilités totales, en tenant compte de l'équiprobabilité des classes :

$$\begin{aligned} \log\left(\mathrm{P}\left(x_{t-L+1:t}\right)\right) &= \log\left(\sum_{i=1}^{p} \mathrm{P}\left(x_{t-L+1:t}|c^i\right)\mathrm{P}\left(c^i\right)\right) \\ &= -\log\left(p\right) + \log\left(\sum_{i=1}^{p} \mathrm{P}\left(x_{t-L+1:t}|c^i\right)\right) \end{aligned} \tag{IV.9}$$

En combinant IV.7, IV.8 et IV.9, l'Équation IV.1 devient :

$$\log\left(\mathrm{P}\left(y_t|x_{t-L+1:t}\right)\right) = \sum_{t'=t-L+1}^{t} \log\left(\mathrm{P}\left(x_{t'}|y_t\right)\right) - \log\left(\sum_{i=1}^{p} \mathrm{P}\left(x_{t-L+1:t}|c^i\right)\right) \tag{IV.10}$$

Le premier terme, qui est le logarithme de la vraisemblance de la fenêtre, est une somme de logarithmes donc ne pose en général pas de problème. En revanche, le deuxième terme implique de calculer la vraisemblance, ce qui conduit aux même problèmes de soupassement. Pour éviter ces problèmes, il est évalué de la manière suivante : on note a_i le logarithme de la vraisemblance de la classe c^i :

$$a_i = \log\left(\mathrm{P}\left(x_{t-L+1:t}|c^i\right)\right) < 0$$

et b le minimum des a_i :

$$b = \min_i a_i$$

On a alors [2] :

$$\begin{aligned}\log\left(\sum_{i=1}^{p}\mathrm{P}\left(x_{t-L+1:t}|c^i\right)\right) &= \log\left(\sum_{i=1}^{p}\exp\left(a_i\right)\right)\\ &= \log\left(\exp\left(b\right)\sum_{i=1}^{p}\exp\left(a_i-b\right)\right)\\ &= b+\log\left(\sum_{i=1}^{p}\exp\left(a_i-b\right)\right)\end{aligned} \tag{IV.11}$$

En général, b ne pose pas de problèmes car c'est un logarithme. La clé de l'astuce est que les termes $a_i - b$ sont en général dans des plages de valeurs pour lesquelles on peut calculer l'exponentielle.

En combinant les équations IV.10 et IV.11, on peut estimer les probabilités *a posteriori* de manière fiable. Grâce à cette formulation, nous pouvons étudier des fenêtres de taille très importante.

Les prochaines parties seront consacrées à l'évaluation des performances du système tBoW.

4. Résultats en reconnaissance d'instances

Dans cette partie, nous évaluons notre système sur la tâche de reconnaissance d'instances ainsi que l'influence des paramètres les plus importants du système sur cette tâche :

- Le type et les paramètres de la signature utilisée.
- La taille de la carte de Kohonen S qui détermine le nombre de clusters $K = S \times S$.
- La taille de la fenêtre temporelle L.

Rappelons que les changements de lieu sont notifiés au système donc l'intégration ne prend en compte que les prototypes apparus après la dernière transition (même avec $L = +\infty$).

4.1. Protocole

Nous utilisons le protocole de reconnaissance d'instances proposé par Ullah et al. [Ull+08] décrit dans la Section A.1.4. Dans ce protocole, on effectue un grand nombre de comparaisons, notamment pour comparer la réponse du système sous différentes conditions de luminosité. Afin de caractériser notre système, nous utilisons le taux de reconnaissance moyen correct et le Taux de Vrais Positifs (TVP) moyen (voir Appendice D). Le lecteur peut se reporter à la Figure IV.7 pour voir un exemple de la présentation détaillée des résultats.

2. Le lecteur reconnaîtra le fameux *log-sum-exp trick*.

5	7	10	15	20	25	30	35	40	45	50

TABLE IV.2.: Valeurs de S utilisées dans la Section IV.4.3.

4.2. Influence de la longueur d'intégration

Pour étudier l'influence de la longueur d'intégration, nous avons répété le protocole pour $L = 1$ puis pour L variant de 10 à 200 par pas de 10 et enfin $L = +\infty$. Ici, la taille de la carte de Kohonen est fixée à $S = 20$.

Les résultats sont présentés dans la Figure IV.3. La figure présente la moyenne sur toutes les répétitions du taux de reconnaissance correcte et du TVP moyen. Nous montrons aussi l'écart-type moyen qui représente la variation des performances pour les différentes répétitions (le TVP de chaque classe est moyenné avant donc cet écart-type ne prend pas en compte la variation du TVP de chaque classe). L'écart-type moyen mesure ainsi la robustesse aux changements dynamiques.

Globalement, accroître L conduit à une amélioration des résultats mais au-delà d'une taille $L = 100$, les effets sont plutôt faibles. L'effet est assez similaire pour la plupart des signatures (on observe un gain d'environ 20%). Notons que CENTRIST a un gain plus important (passant de 42 à 79%). L'écart-type augmente légèrement avec L (généralement de moins de 3%).

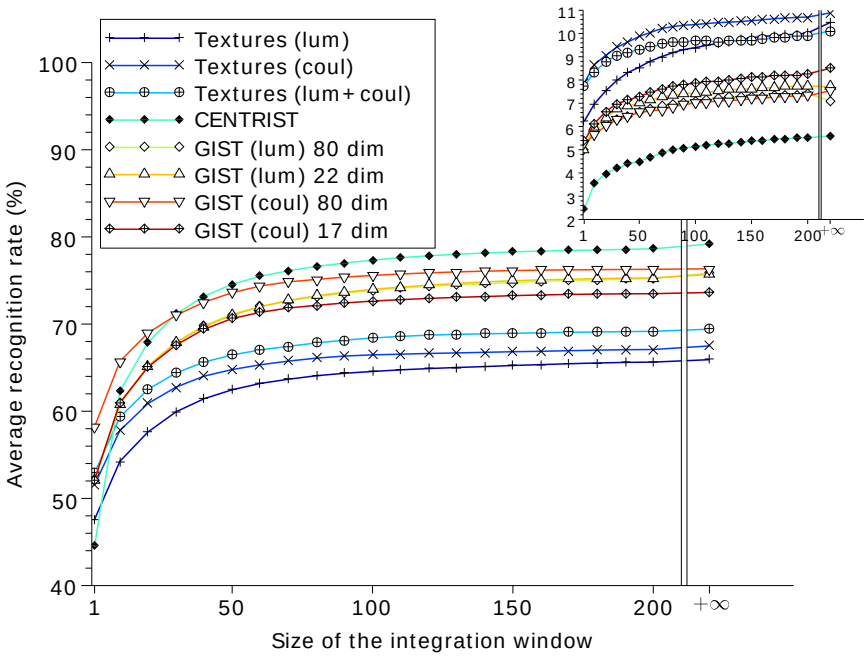
Ces résultats montrent que la méthode d'intégration proposée permet d'améliorer les performances. Ils confirment l'apport du travail de [Gui09] en testant le système sur une base de données incluant plus de variations.

4.3. Influence de la taille de la carte de Kohonen

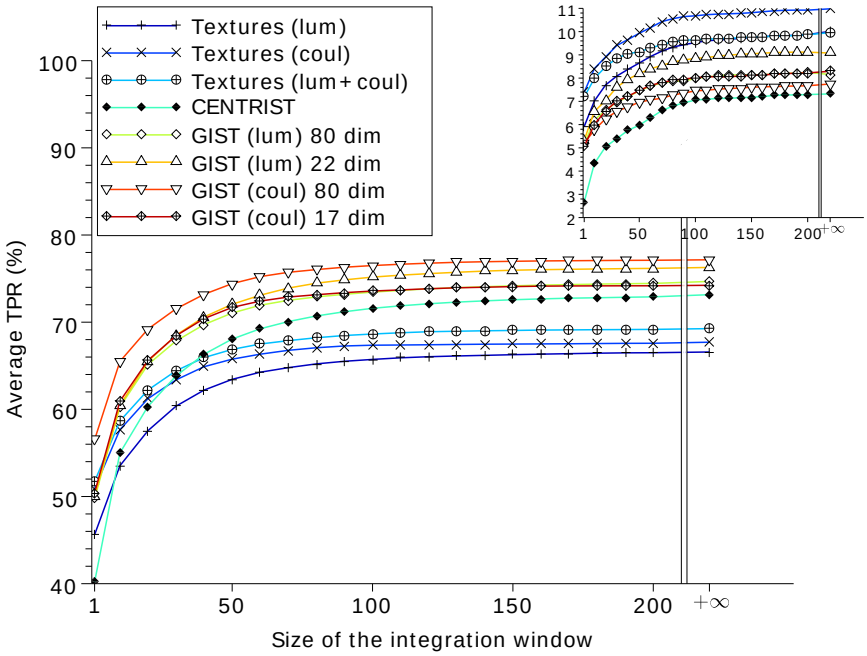
Pour étudier l'influence de la taille de la carte de Kohonen, nous avons répété le protocole avec différentes valeurs de S (voir Table IV.2). Ici, la longueur d'intégration est $L = +\infty$.

Les résultats sont présentés dans la Figure IV.4. Globalement, accroître S conduit à une amélioration des résultats mais au-delà d'une taille $S = 25$, les effets sont plutôt faibles. Cependant, contrairement à la taille de la fenêtre d'intégration, l'effet dépend des signatures :

- il est assez faible pour les signatures basées sur les fonctions de texture (passage de 75% à 85 %). L'écart-type tend à augmenter.
- il est très important pour CENTRIST (le taux de reconnaissance passant de 60% à 90%). Cela peut s'expliquer par le fait que CENTRIST possède 256 dimensions : la caractérisation d'un espace de grande dimension nécessite beaucoup plus de clusters. L'écart-type diminue fortement.

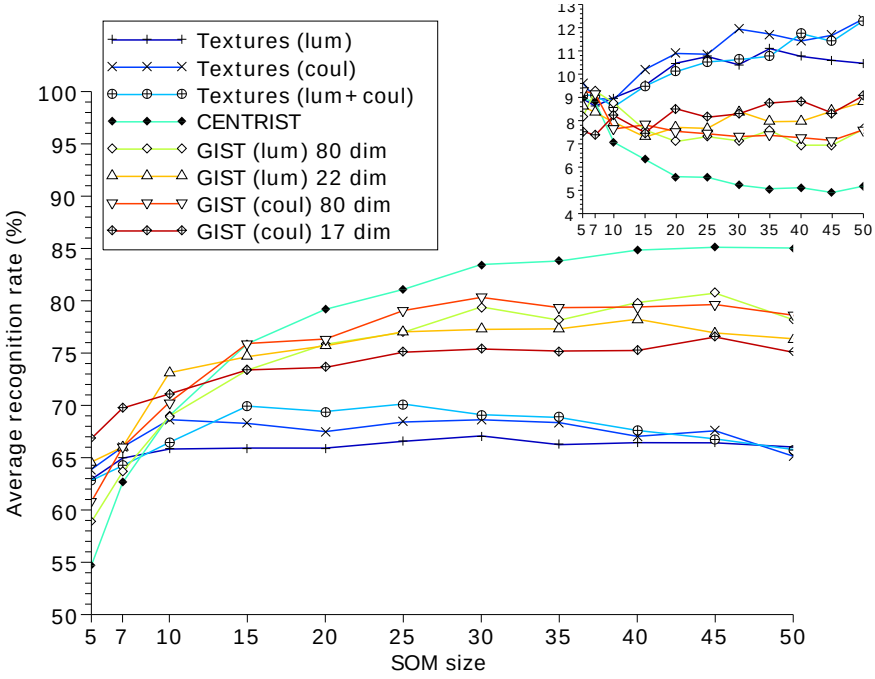


(a) Taux de reconnaissance moyen

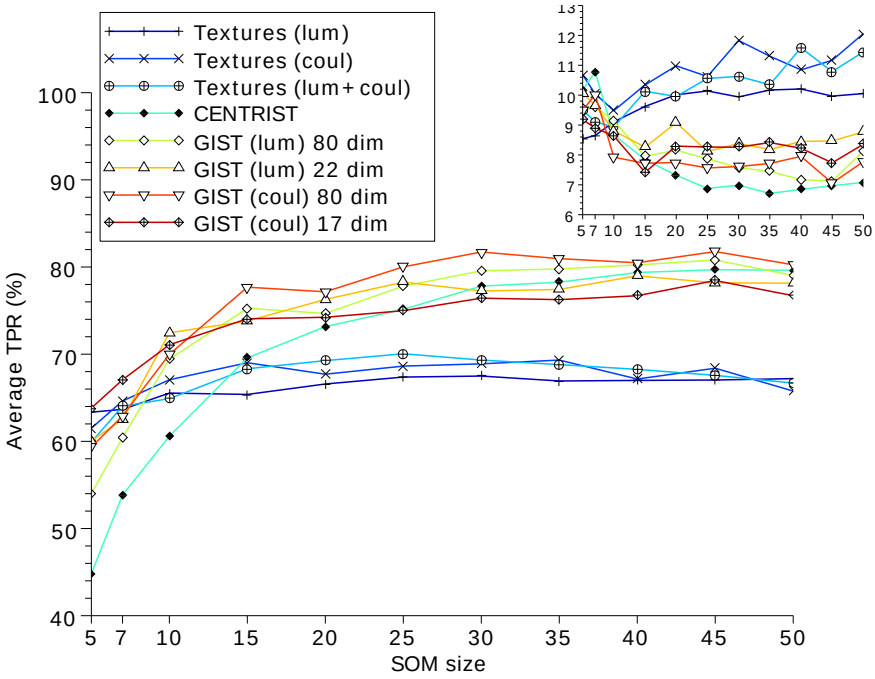


(b) TVP

FIGURE IV.3.: Influence de L en reconnaissance d'instances : (a) sur le taux de reconnaissance moyen, (b) sur le TVP moyen. Les encarts donnent l'évolution de l'écart-type moyen.



(a) Taux de reconnaissance moyen



(b) TVP

FIGURE IV.4.: Influence de S en reconnaissance d'instances : (a) sur le taux de reconnaissance moyen, (b) sur le TVP moyen. Les encarts donnent l'évolution de l'écart-type moyen.

– concernant GIST, le taux de reconnaissance passe de 70% à plus de 90%. L'écart-type diminue légèrement.

L'effet de la taille de la SOM semble lié à la dimensionnalité du descripteur : les signatures de faible dimension (les signatures calculées grâce aux fonctions de textures) voient leur performance augmenter faiblement, alors que les signatures de grande dimension (comme CENTRIST) voient leur performance augmenter fortement.

Ces résultats sont assez surprenants pour deux raisons. Tout d'abord, on aurait pu penser que les SOM de taille moyenne permettent de réaliser un bon compromis entre la variabilité et la confusion perceptuelle. En effet, des SOM de petite taille regroupent beaucoup d'images sous un même prototype, ce qui risque d'introduire une grande confusion. *A contrario*, dans une SOM trop grande, les prototypes sont trop spécialisés. De plus, dans ce protocole, le nombre d'images par séquence d'apprentissage est compris entre 646 et 2 942. Pour les SOM de grande taille, une faible proportion de cellules va être activée durant l'apprentissage, ce qui rend plus difficile une estimation correcte de $\mathrm{P}\left(x_t|y_t\right)$.

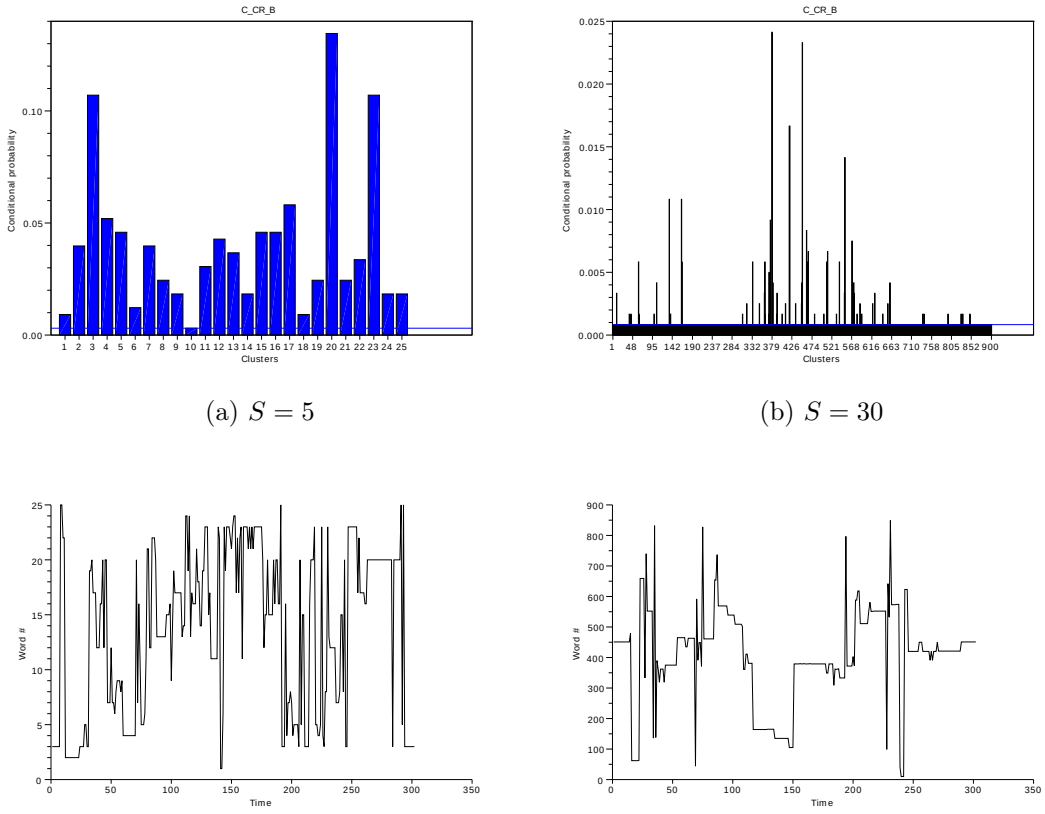
La Figure IV.5 montre un exemple des modèles obtenus pour différentes valeurs de S et les séquences dont ils sont issus. On voit que les séquences d'entraînements avec des SOM de grande taille sont plus stables (*i.e.* le prototype change moins souvent).

Cela nous conduit à l'hypothèse que les mots visuels pour les SOM de plus grande taille sont plus robustes (des images proches sont associées au même mot). Pour vérifier cela, nous avons calculé la distance moyenne entre le modèle (*i.e.* la distribution des prototypes dans la séquence d'entraînement) et la distribution des prototypes dans les séquences de reconnaissance. La distance utilisée est la distance d'intersection d_{inter} (voir [SB91]). La Table IV.3 donne les résultats : on voit que pour les SOM de taille importante, les distributions en apprentissage et en reconnaissance sont plus proches. Autrement dit, les prototypes trouvés dans la séquence d'apprentissage sont plus souvent trouvés dans les séquences de reconnaissance (qui contiennent des images issues d'une autre condition de luminosité).

Enfin, la Figure IV.6 montre un échantillon des images associées à la cellule la plus probable des modèles de la Figure IV.5. On constate qu'avec des SOM de taille faible, des images d'autres lieux sont associées au même prototype.

Ces observations nous permettent de dire que les SOM de grande taille sont plus intéressantes car elles permettent de trouver des images très représentatives du lieu. Ces images deviennent donc très discriminantes. Ces résultats ne remettent pas en cause fondamentalement l'approche de Guillaume, Denquive et Tarroux [GDT05] (voir section IV.2) car ces cellules peuvent aussi être représentatives d'autres lieux.

Dans l'ensemble, ces résultats confirment qu'un grand dictionnaire est intéressant pour les méthodes basées sur les mots visuels, comme cela a été souligné dans la Section II.3.1. Il est nécessaire d'utiliser des SOM de grande taille pour obtenir un découpage pertinent de



(a) $S = 5$ (b) $S = 30$

(c) $S = 5$ (d) $S = 30$

FIGURE IV.5.: Influence de S sur l'apprentissage. (a)-(b) : Modèles de la classe « couloir » obtenus pour différentes valeurs de S avec le descripteur GIST. Le graphique montre la probabilité d'apparition de chaque prototype. La ligne horizontale bleue indique la probabilité par défaut pour les prototypes non-vus (voir Équation IV.5). Les modèles sont issus de l'apprentissage de la classe couloir réalisé sur la séquence $(SAARBRUECKEN, B, STD, NIGHT, 1)$. Il y a 302 images dans l'échantillon d'apprentissage. (c)-(d) : Prototypes observés dans la classe « couloir » au cours du temps que la même séquence. L'axe horizontal est le temps et l'axe vertical le numéro du prototype observé à cet instant.

S	d_{inter}
5	0,700
10	0,651
20	0,669
30	0,729
40	0,783
50	0,823

TABLE IV.3.: Distance d'intersection moyenne entre les distribution d'apprentissage et les distributions en reconnaissance (pour la tâche de reconnaissance d'instances). Plus la distance d'intersection est proche de 1, plus les distributions sont similaires.

(a) $S = 5$

(b) $S = 30$

FIGURE IV.6.: Images associées à la cellule la plus vraisemblable de la classe couloir.

Signature	Variation du Taux de reconnaissance	Variation du TVP moyen
Texture (lum)	-15,74 %	-15,39 %
Texture (coul)	-17,17 %	-16,51 %
Texture (lum+coul)	-16,86 %	-16,10 %
CENTRIST	**-6,91 %**	**-8,51 %**
GIST (lum) 80 dim	-11,54 %	-11,92 %
GIST (lum) 22 dim	-12,12 %	-11,49 %
GIST (coul) 80 dim	-12,64 %	-11,53 %
GIST (coul) 17 dim	-13,17 %	-12,32 %

TABLE IV.4.: Robustesse aux variations de luminance en reconnaissance d'instances : différence du taux de reconnaissance et du TVP obtenus en utilisant uniquement les conditions d'illumination congruentes et dans le cas général. Plus la valeur absolue est petite, plus la robustesse est grande.

l'espace de description des images. Au-delà d'une certaine taille, qui dépend de la dimension du descripteur, rajouter des cellules ne conduit pas à une amélioration des performances.

4.4. Robustesse aux variations de luminosité

Nous étudions maintenant la robustesse de notre système au changement de conditions de luminosité. Pour cela, nous comparons les performances entre le cas où les conditions de luminosité sont les mêmes en apprentissage et en reconnaissance (on dit qu'elles sont congruentes) et le cas général. Ici, la taille du Kohonen est fixée à $S = 20$ et la taille de la fenêtre d'intégration à $L = +\infty$.

Les résultats sont présentés dans la Table IV.4. Globalement, CENTRIST semble le descripteur le plus robuste. Rappelons que CENTRIST est basé sur une étape préalable de détection des contours (voir Section I.2.2.2) à l'aide d'un filtre de Sobel, ce qui explique qu'il soit moins sensible aux variations d'intensité. De plus, la Census Transform (CT) est robuste aux changements d'illumination. Les descripteurs basés sur les fonctions de texture sont les moins robustes, ce qui était attendu car ces signatures travaillent directement avec les valeurs

des pixels.

4.5. Comparaison avec l'état de l'art et conclusion

En ce qui concerne les signatures, GIST et CENTRIST donnent presque toujours des résultats supérieurs aux signatures basées sur les fonctions de texture tout en étant plus robustes. Le meilleur taux de reconnaissance correcte est obtenu avec CENTRIST. En revanche, si l'on regarde le TVP, le descripteur GIST donne de meilleurs résultats.

Les différentes variantes du descripteur GIST n'ont pas de différence très importante. Ce fait, en contradiction avec le rôle des couleurs dans le processus de reconnaissance d'instances (voir section III.4), est peut-être lié à la prise en compte de conditions de luminosité pour lesquelles les couleurs ne sont pas utiles. De plus, l'utilisation d'un nombre de dimensions plus faible semble avoir peu d'impact sur les performances (mais il est un peu plus important pour les signatures calculées sur les canaux de couleurs).

Enfin, notons que nous n'avons pas étudié les interactions entre les différents facteurs. On peut supposer que plus S augmente, plus l'effet de L est important. En effet, si S est grand, on a besoin de plus de prototypes pour estimer correctement la ressemblance entre la distribution d'apprentissage et la distribution de reconnaissance.

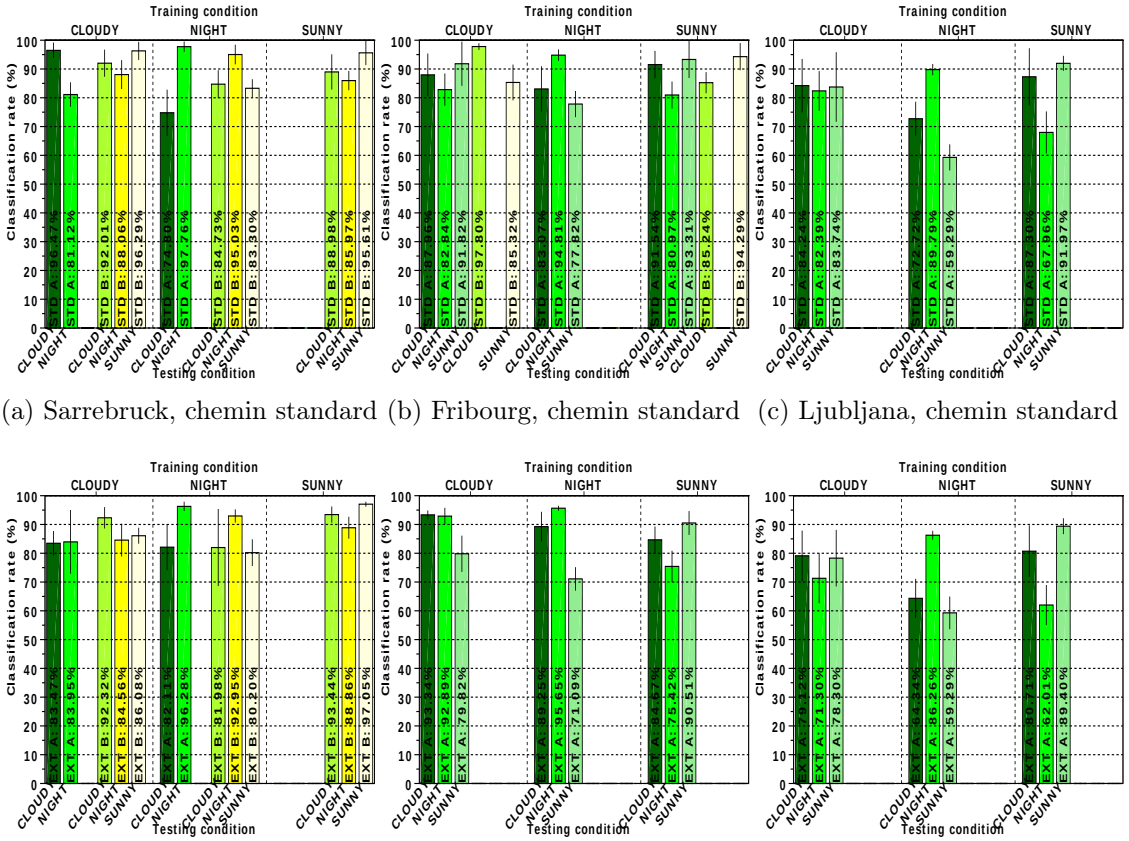
Afin de comparer nos travaux à [Ull+08], nous présentons les résultats des deux systèmes ayant les meilleures performances :

- la signature CENTRIST avec $S = 50$ et $L = +\infty$ (voir Figure IV.7)
- la signature GIST (calculée sur les canaux de couleurs et projetée sur 80 dimensions) avec $S = 30$ et $L = +\infty$ (voir Figure IV.8)

La comparaison a été effectuée avec le test t de Student. Dans le cas de la comparaison de plusieurs classificateurs sur un même jeu de test, on recommande en général le test de Friedman (voir [CM10, chap. 3]) mais celui-ci est applicable quand on fait de la validation croisée, ce qui n'est pas le cas ici. Les résultats montrent que :

- le système basé sur CENTRIST a une moyenne supérieure à celui de [Ull+08] ($p < 1\%$).
- il n'y a pas de différence significative dans la moyenne entre le système basé sur GIST et celui de [Ull+08] (ce qui signifie que les performances sont au même niveau). Cela est dû aux mauvais résultats dans le laboratoire de Ljubljana (voir les figures IV.8c et IV.8f), notamment la nuit. Ces séquences ont été obtenues dans des conditions de luminosité très faible, ce qui explique sans doute les faible performances. Si l'on ne tient pas compte de ces séquences, le système basé sur GIST a une moyenne supérieure à celui de [Ull+08] ($p < 1\%$).

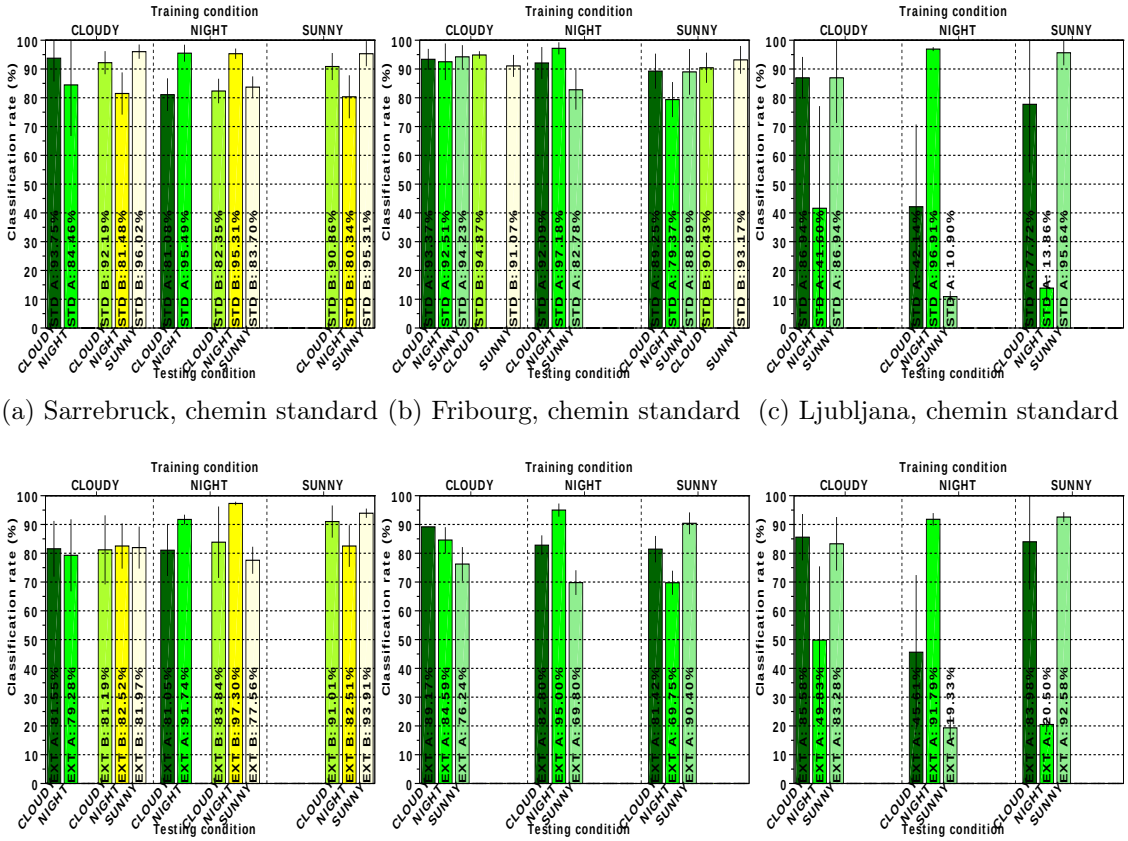
Globalement, on peut donc dire que le système tBoW est meilleur que l'état de l'art en ce qui concerne la tâche de reconnaissance d'instances de lieux.



(a) Sarrebruck, chemin standard (b) Fribourg, chemin standard (c) Ljubljana, chemin standard

(d) Sarrebruck, chemin étendu (e) Fribourg, chemin étendu (f) Ljubljana, chemin étendu

FIGURE IV.7.: Résultats en reconnaissance d'instances obtenus avec le descripteur CENTRIST. Les barres d'incertitude représentent l'écart-type. Les barres manquantes correspondent aux conditions manquantes dans la base de données. Le laboratoire et la partie utilisés sont indiqués dans la barre. La ligne du haut montre les séquences issues du chemin standard et celle du bas, les séquences issues du chemin étendu. La condition d'illumination utilisée en apprentissage est montrée en haut de chaque figure. La condition d'illumination utilisée en reconnaissance est montrée sous chaque barre.



(a) Sarrebruck, chemin standard (b) Fribourg, chemin standard (c) Ljubljana, chemin standard

(d) Sarrebruck, chemin étendu (e) Fribourg, chemin étendu (f) Ljubljana, chemin étendu

FIGURE IV.8.: Résultats en reconnaissance d'instances obtenus avec le descripteur GIST. Les barres d'incertitude représentent l'écart-type. Les barres manquantes correspondent aux conditions manquantes dans la base de données. Le laboratoire et la partie utilisés sont indiqués dans la barre. La ligne du haut montre les séquences issues du chemin standard et celle du bas, les séquences issues du chemin étendu. La condition d'illumination utilisée en apprentissage est montrée en haut de chaque figure. La condition d'illumination utilisée en reconnaissance est montrée sous chaque barre.

5. Résultats en catégorisation

Dans cette partie, nous cherchons à évaluer notre système sur la tâche de catégorisation de lieux sémantiques ainsi que l'influence des paramètres sur cette tâche.

5.1. Protocole

Nous utilisons le protocole de catégorisation proposé par [Ull+08] décrit dans la Section A.1.5. Le protocole utilise un sous-ensemble de la base COLD qui comprend quatre classes. L'apprentissage est réalisé avec une séquence extraite de deux laboratoires et la reconnaissance s'effectue sur une séquence issue du troisième laboratoire. La condition de luminosité utilisée est la même dans toutes les séquences. Rappelons que les changements de catégorie sont notifiés au système.

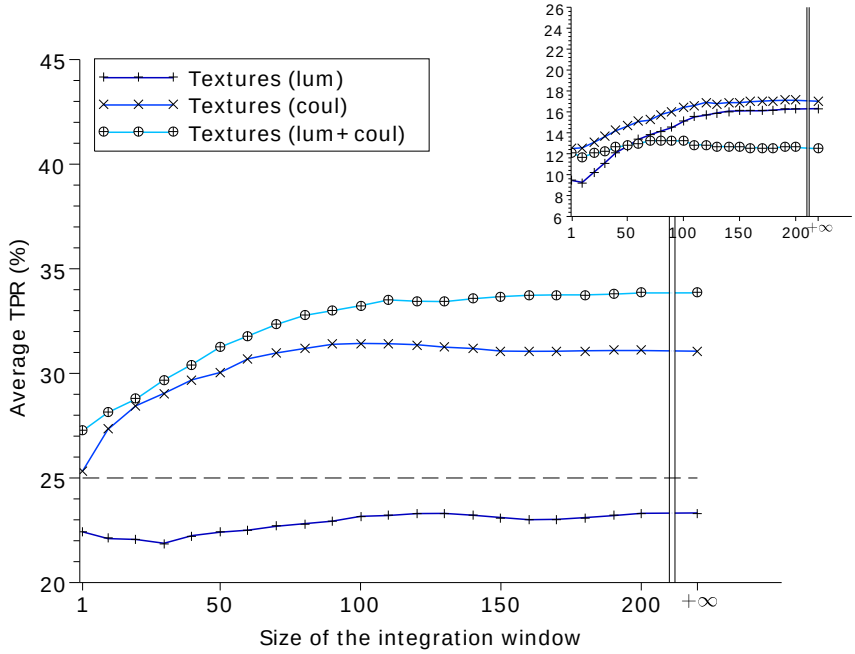
5.2. Influence de la longueur d'intégration

Pour étudier l'influence de la longueur d'intégration, nous avons répété le protocole pour $L = 1$ puis pour L variant de 10 à 200 par pas de 10 et enfin $L = +\infty$. Ici, la taille de la carte de Kohonen est fixée à $S = 20$. Afin de caractériser notre système, nous utilisons le TVP moyen (voir Appendice D). Le taux de reconnaissance correcte moyen est peu informatif car le taux de reconnaissance des classes varie beaucoup.

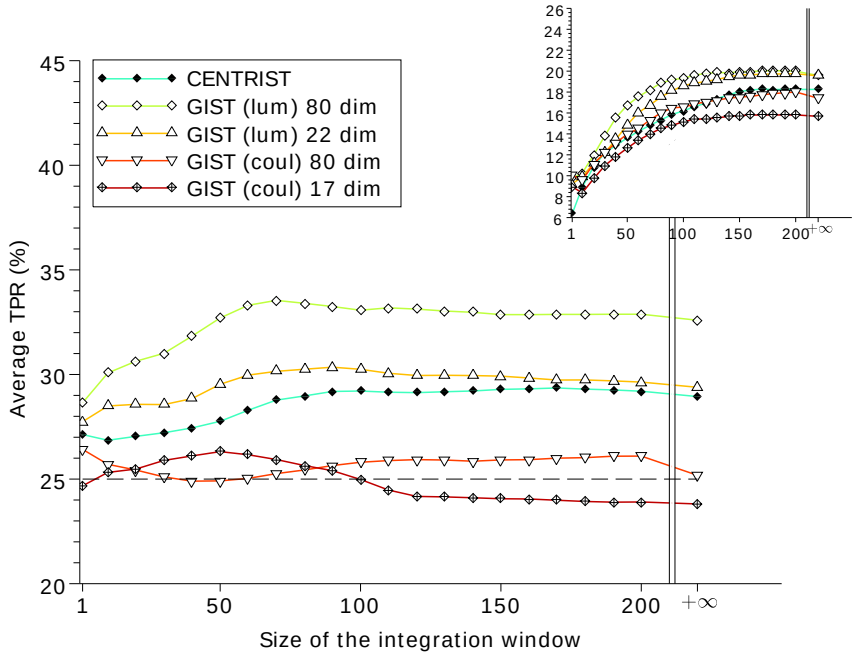
Les résultats sont présentés dans la Figure IV.9. La figure présente la moyenne sur toutes les répétitions du TVP moyen. Nous montrons aussi l'écart-type moyen qui représente la variation des performances pour les différentes répétitions (comme dans la section précédente, cet écart-type ne prend pas en compte la variation du TVP entre les classes).

L'effet de L est faible mais positif pour les signatures CENTRIST et les fonctions de texture sur le canal de luminance. Il y a un effet positif marqué pour les signatures calculées sur les fonctions de texture utilisant la couleur (voir figure IV.9a). Cependant (voir Figure IV.10), cela est principalement dû à de très forts taux de reconnaissance sur certaines pièces alors que d'autres, en particulier le couloir, sont très mal reconnues. Les signatures GIST et CENTRIST ne présentent pas de variations aussi extrêmes.

Notons que les signatures GIST calculées sur le canal de luminance ont un avantage clair par rapport à celles calculées sur le canal de couleur (voir figure IV.9b). L'augmentation de L jusqu'à environ 60 augmente les performances pour les signatures GIST calculées sur le canal de luminance.



(a) Signatures basées sur les fonctions de texture.



(b) CENTRIST et GIST.

FIGURE IV.9.: Influence de L sur le TVP moyen en catégorisation. La ligne horizontale en pointillés indique la ligne de chance. L'encart représente l'évolution de l'écart-type moyen.

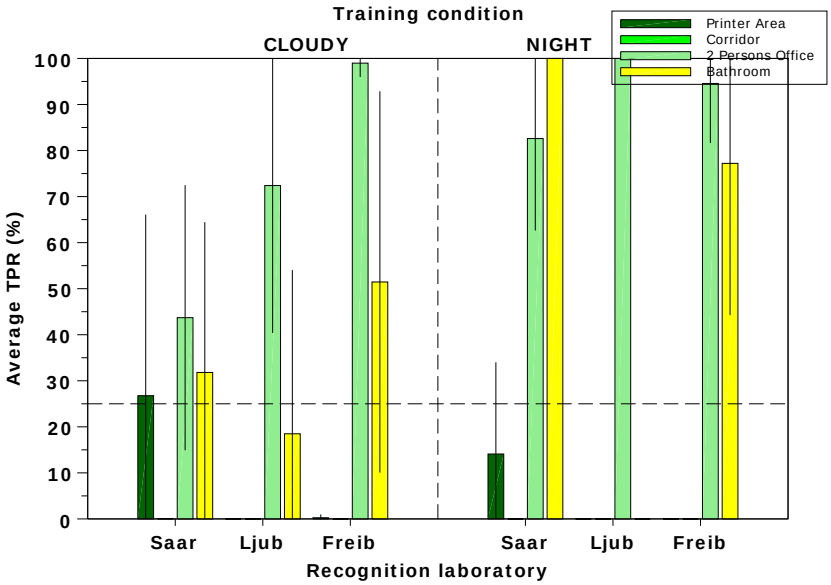


FIGURE IV.10.: Résultats obtenus en catégorisation avec les fonctions de texture ($L = +\infty$, $S = 20$) : certaines classes sont très bien reconnues et d'autres ont un taux de reconnaissance nul. Les barres d'erreur représentent l'écart-type. La condition d'illumination utilisée pour l'entraînement et la reconnaissance est montrée sur l'axe horizontal haut. Le laboratoire utilisé pour la reconnaissance est indiqué en dessous. La ligne horizontale en pointillés indique la ligne de chance.

5.3. Influence de la taille de la carte de Kohonen

Pour étudier l'influence de la taille de la SOM, nous avons répété le protocole pour les valeurs de S données dans la Table IV.2. Ici, la longueur d'intégration est $L = +\infty$. Ici aussi, nous utilisons le TVP moyen.

Les résultats sont présentés dans la Figure IV.11. L'effet de S dépend beaucoup de la signature et de sa dimensionnalité :

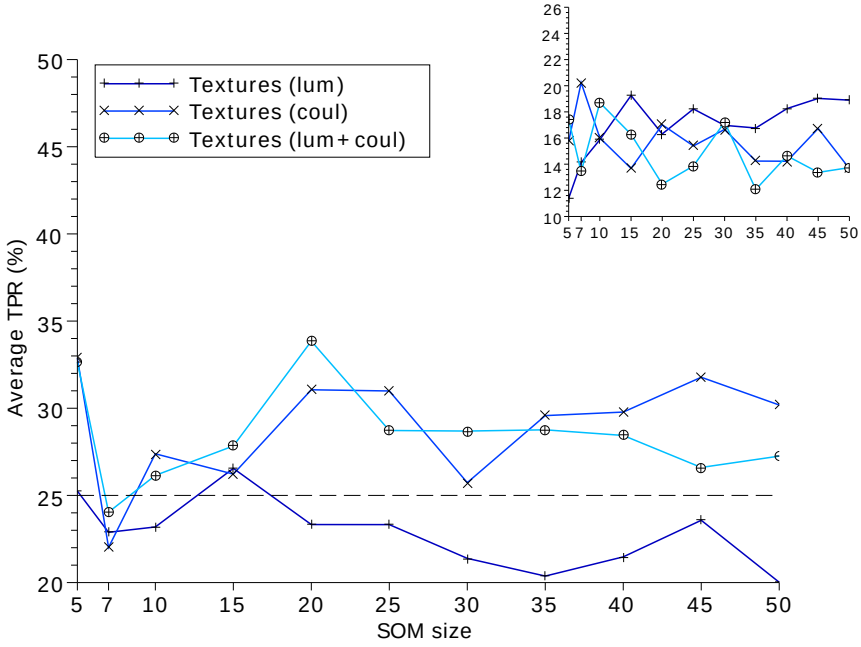
- Il est globalement négatif pour les fonctions de texture basées sur la luminance seule, cette signature montrant des performances assez faibles.
- Pour les fonctions de texture basées sur les couleurs, les performances augmentent jusqu'à $S = 20$ puis restent stables ou diminuent légèrement.
- Pour CENTRIST, les performances sont bonnes pour $S = 5$, chutent pour $S = 7$ et remontent ensuite (avec une diminution à $S = 20$).
- Pour les signatures basées sur GIST, on observe le même profil que pour CENTRIST : les SOM de taille faible et les SOM de taille importante donnent de meilleures performances que les SOM de taille intermédiaire. Il y a toutefois des variations assez importantes d'une taille à l'autre. Les variantes utilisant les canaux de couleur ont des performances généralement moins bonnes que les variantes utilisant uniquement la luminance.

Comme dans la Section IV.4.3, nous avons calculé la distance d'intersection moyenne entre la distribution d'apprentissage et la distribution en reconnaissance. La Table IV.5 donne les résultats pour le descripteur GIST (calculé sur le canal de luminance). La distance d'intersection d_{inter} suit la même évolution que les performances.

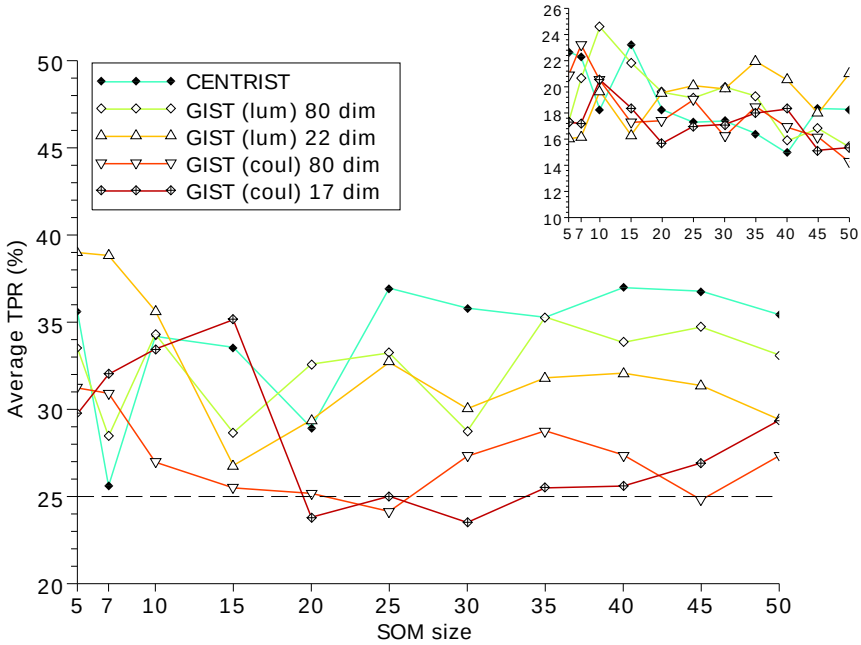
On peut donc supposer que les prototypes ne sont pas robustes à la variabilité environnementale. Plus de travaux sont nécessaires pour savoir si cela est dû aux signatures ou aux cartes de Kohonen.

5.4. Comparaison avec l'état de l'art et conclusion

Globalement, les signatures GIST calculées sur le canal de luminance et CENTRIST semblent les plus intéressantes même si les signatures calculées à partir des fonctions de texture peuvent être très efficaces pour certaines pièces. Concernant la taille de la fenêtre d'intégration, on voit qu'une fenêtre temporelle assez grande (entre 50 et 100) donne les meilleurs résultats. Contrairement au cas de la reconnaissance d'instances, les cartes de Kohonen très grandes ($S > 25$) n'apportent pas d'avantage.



(a) Signatures basées sur les fonctions de texture.



(b) CENTRIST et GIST.

FIGURE IV.11.: Influence de S sur le TVP moyen en catégorisation. La ligne horizontale en pointillés indique la ligne de chance. L'encart représente l'évolution de l'écart-type moyen.

S	d_{inter}
5	0,72
10	0,45
20	0,34
30	0,41
40	0,50
50	0,58

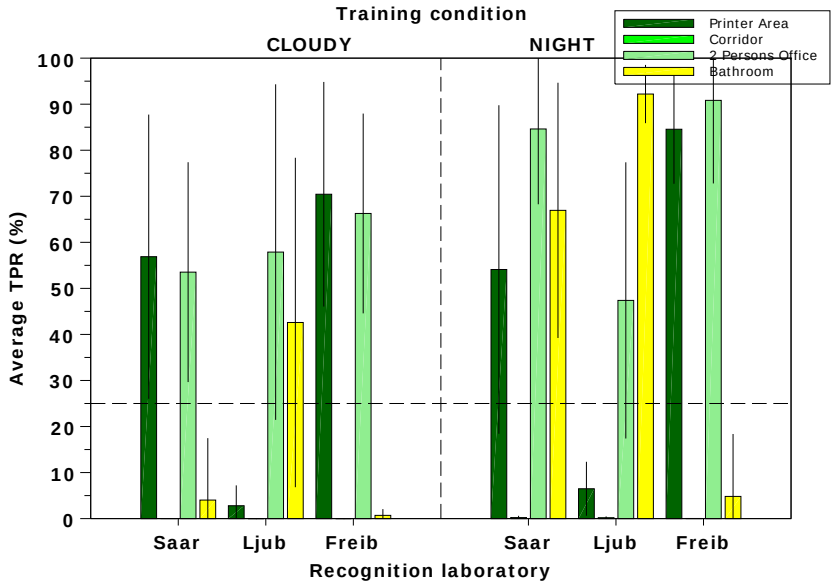
TABLE IV.5.: Distance d'intersection moyenne entre les distribution d'apprentissage et les distributions en reconnaissance. Plus la distance d'intersection est proche de 1, plus les distributions sont similaires.

5.4.1. Comparaison sur la base COLD

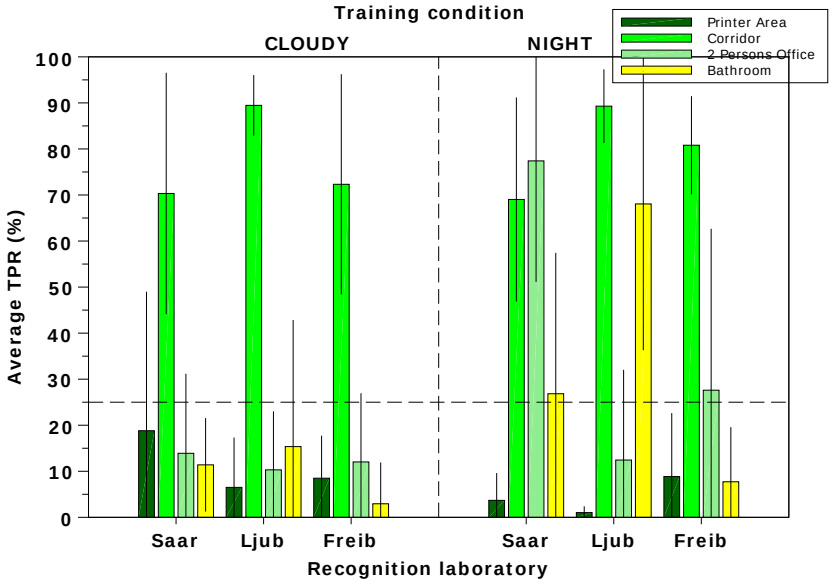
Afin de comparer nos travaux à ceux de ULLAH et al. [Ull+08] (voir Figure III.13), nous présentons les résultats détaillés de deux systèmes :

- la signature CENTRIST avec $S = 40$ et $L = +\infty$ (voir figure IV.12a)
- la signature GIST (calculée sur le canal de luminance et projetée sur 80 dimensions) avec $S = 5$ et $L = +\infty$ (voir figure IV.12b)

Il y a trop peu de données dans ce protocole pour pouvoir effectuer un test statistique. Néanmoins, le TVP moyen est de 33% dans le cas de GIST et de 37% dans le cas de CENTRIST (il est d'environ 28% dans [Ull+08], voir Figure III.13, page 137). De plus, dans [Ull+08], seule la classe couloir dépasse le seuil de chance, les autres restant largement en-dessous alors que notre système donne de bons résultats sur plusieurs catégories. Le descripteur GIST montre de bonnes performances dans le couloir et parfois les bureaux et les toilettes. Le couloir est, en général, caractérisé par des orientations très marquées dues aux murs et à l'allongement. Il n'est donc pas étonnant que GIST donne de bonnes performances sur cette catégorie. Le descripteur CENTRIST donne de bons résultats dans la salle de reprographie et les bureaux et, dans certains cas, les toilettes. Ces catégories sont plus riches visuellement que le couloir (présence de nombreux objets), ce qui explique les bons résultats de CENTRIST.



(a) CENTRIST



(b) GIST

FIGURE IV.12.: Exemples de résultats en catégorisation avec le système tBoW : (a) avec le descripteur CENTRIST (b) avec le descripteur GIST. Les barres d'erreur représentent l'écart-type. La condition d'illumination utilisée pour l'entraînement et la reconnaissance est montrée sur l'axe horizontal haut. Le laboratoire utilisé pour la reconnaissance est indiqué en dessous. La ligne horizontale en pointillés indique la ligne de chance.

	Categories					
Home	**Bed**	**Bath**	**Kitchen**	**Living**	**Dining**	**Average**
1	20,17 %	88,16 %	41,29 %	9,76 %	0,00 %	31,87 %
2	11,55 %	65,90 %	99,19 %	1,13 %	80,92 %	51,74 %
3	9,11 %	96,98 %	24,56 %	1,58 %	71,17 %	40,68 %
4	67,83 %	89,86 %	87,89 %	27,45 %	1,48 %	54,90 %
5	30,71 %	99,86 %	71,16 %	0,00 %	0,00 %	40,34 %
6	51,20 %	62,69 %	27,45 %	95,59 %	0,00 %	47,39 %
Average	31,76 %	83,91 %	58,59 %	22,58 %	25,59 %	44,49 %

TABLE IV.6.: Performances obtenues sur la base VPC pour le descripteur CENTRIST ($S = 40$, $L = +\infty$).

5.4.2. Comparaison sur la base VPC

Afin de comparer nos résultats à ceux de [WCR09 ; Ran10], nous avons testé le système sur la base VPC en reproduisant le protocole décrit dans la Section A.2.3. Ce protocole utilise cinq catégories présentes dans toutes les maisons. Le protocole est une variante de la validation croisée appelée *leave-one-out*.

Nous avons utilisé les paramètres qui ont donné les meilleurs résultats sur la base COLD. Il n'y a cependant pas de garantie qu'ils donnent les meilleurs résultats sur ce protocole. En particulier, les cartes de Kohonen sur la base VPC ont été entraînées avec moins d'images. Il est donc possible que des cartes plus petites donnent de meilleurs résultats. Les Table IV.6 et IV.7 donnent le TVP par classe.

Ces résultats sont de l'ordre des performances reportées dans WU, CHRISTENSEN et REHG [WCR09] et RANGANATHAN [Ran10] (voir Table III.1), bien que légèrement inférieurs. Nous avons comparé nôtre système et celui de [WCR09] avec un test t : avec un niveau de confiance de 95%, les différences sont significatives.

Les résultats obtenus avec CENTRIST suivent la même tendance que [WCR09] (qui utilisait aussi ce descripteur) avec de bonnes performances pour les catégories « cuisine » et « salle de bain » et des performances beaucoup moins bonnes pour les catégories « salon » et « salle

	Categories					
Home	**Bed**	**Bath**	**Kitchen**	**Living**	**Dining**	**Average**
1	37,50 %	84,07 %	35,06 %	82,93 %	39,02 %	55,72 %
2	23,36 %	38,67 %	72,67 %	0,00 %	38,52 %	34,64 %
3	27,46 %	91,55 %	80,18 %	51,26 %	0,00 %	50,09 %
4	53,91 %	32,31 %	69,51 %	54,09 %	7,21 %	43,41 %
5	36,10 %	23,32 %	29,08 %	94,49 %	0,00 %	36,60 %
6	36,85 %	77,03 %	4,68 %	99,83 %	0,89 %	43,86 %
Average	35,87 %	57,83 %	48,53 %	63,77 %	14,27 %	44,05 %

TABLE IV.7.: Performances obtenues sur la base VPC pour le descripteur GIST calculé sur le canal de luminance ($S = 5$, $L = +\infty$).

à manger ». Là aussi, le descripteur GIST montre de bonnes performances sur d'autres catégories (« salon » et « cuisine »). Le test t indique qu'il n'y a pas de différence significative entre les résultats obtenus avec GIST et ceux obtenus avec CENTRIST.

5.5. Conclusion

Nous avons présenté les résultats de notre système pour deux protocoles différents de catégorisation. À notre connaissance, c'est la première fois qu'un système est évalué sur plusieurs bases. Les résultats obtenus sur COLD sont meilleurs que l'état de l'art. Les résultats obtenus sur VPC sont légèrement inférieurs à l'état de l'art. Compte-tenu du fait que nous avons choisi les paramètres à partir des performances sur la base COLD, cela est plutôt encourageant.

Cependant, dans les deux cas, on note que les performances varient beaucoup d'une catégorie à l'autre. De plus, il n'y a pas de catégorie plus facile que d'autre : les résultats pour une catégorie peuvent être très bons pour un descripteur et très mauvais pour un autre. La tâche de catégorisation est donc toujours ouverte et plus de travaux sont nécessaires, notamment la définition de descripteurs présentant de bonnes performances pour l'ensemble des classes.

6. Conclusion

Ce chapitre a présenté et évalué en profondeur le système tBoW. Ce système est basé sur des vues prototypiques caractérisant l'image dans son ensemble, obtenues par un algorithme de quantification vectorielle et un modèle d'apprentissage génératif et incrémental. En phase de reconnaissance, l'utilisation d'une fenêtre de taille finie permet d'intégrer plusieurs images.

Le système a montré des performances supérieures au niveau de l'état de l'art en reconnaissance d'instances et en catégorisation. De part l'utilisation d'une signature globale et d'un modèle de type BoW, notre système est relativement simple.

Nous avons montré que les performances sont améliorées lorsqu'on augmente la taille de la fenêtre d'intégration temporelle. Nous avons aussi montré que l'utilisation de nombreux prototypes permettait de sélectionner des images pertinentes pour la reconnaissance d'instances.

Ce chapitre nous a aussi permis de comparer différentes signatures globales utilisées en reconnaissance de lieux sémantiques sur une même base. Une telle comparaison n'était pas disponible dans la littérature. Il ressort que les signatures GIST et CENTRIST montrent de bonnes performances, notamment une meilleure robustesse aux changements de conditions d'illumination. Concernant GIST, l'utilisation des canaux de couleurs montre peu d'intérêt en reconnaissance d'instances et diminue nettement les performances en catégorisation. On peut donc se contenter d'utiliser uniquement le canal de luminance. De plus, l'utilisation d'un nombre plus réduit de composantes principales (obtenu par la règle des 95% de variance) ne semble pas apporter de bénéfice.

Rappelons que la principale limitation du système est qu'il nécessite un mécanisme externe pour détecter les transitions. Le prochain chapitre présentera les travaux que nous avons réalisés pour corriger ce problème.

Bibliographie

[CM10] A. Cornuéjols et L. Miclet. *Apprentissage artificiel : concepts et algorithmes*. 2e éd. Paris : Eyrolles, juin 2010, p. 804 (cf. p. 163, 174).

[GDT05] H. Guillaume, N. Denquive et P. Tarroux. « Contextual priming for artificial visual perception ». Dans : *European Symposium on Artificial Neural Networks (ESANN 2005)*. Bruges, Belgium, 2005, p. 545–550 (cf. p. 154, 170).

[Gol91] D. Goldberg. « What every computer scientist should know about floating-point arithmetic ». Dans : *ACM Comput. Surv.* 23 (1 mar. 1991), p. 5–48. ISSN : 0360-0300 (cf. p. 164).

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 153–156, 158, 159, 164, 167).

[HSD73] R. M. Haralick, K. Shanmugam et I. Dinstein. « Textural Features for Image Classification ». Dans : *IEEE Transactions on Systems, Man and Cybernetics* 3.6 (nov. 1973), p. 610–621 (cf. p. 154).

[Kai74] H. Kaiser. « An index of factorial simplicity ». Dans : *Psychometrika* 39 (1 1974). 10.1007/BF02291575, p. 31–36. ISSN : 0033-3123 (cf. p. 160).

[Koh89] T. Kohonen. *Self-organization and associative memory*. 3e éd. Berlin : Springer-Verlag, 1989 (cf. p. 155).

[OT01] A. Oliva et A. Torralba. « Modeling the shape of the scene : A holistic representation of the spatial envelope ». Dans : *International Journal of Computer Vision* 42.3 (2001), p. 145–175 (cf. p. 155).

[Pro+10] A. Pronobis et al. « Multi-modal semantic place classification ». Dans : *The International Journal of Robotics Research* 29.2-3 (2010), p. 298–320 (cf. p. 164).

[Ran10] A. Ranganathan. « PLISS : Detecting and Labeling Places using Online Change-point Detection ». Dans : *Proceedings of the 2010 Robotics : Science and Systems Conference (RSS 2010)*. 2010 (cf. p. 164, 184).

[RN99] D. Radhakrishnan et I. Nourbakhsh. « Topological robot localization by training a vision-based transition detector ». Dans : *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*. T. 1. IEEE. 1999, p. 468–473 (cf. p. 164).

[SB91] M. J. Swain et D. H. Ballard. « Color indexing ». Dans : *International Journal of Computer Vision* 7 (1991), 11–32 (cf. p. 170).

[Tor+03] A. Torralba et al. *Context-based vision system for place and object recognition*. Rap. tech. Cambridge, MA, 2003 (cf. p. 159, 160).

[Ull+08] M. M. Ullah et al. « Towards Robust Place Recognition for Robot Localization ». Dans : *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008)*. Pasadena, USA, mai 2008 (cf. p. 166, 174, 177, 182).

[VA02] J. Vesanto et E. Alhoniemi. « Clustering of the self-organizing map ». Dans : *Neural Networks, IEEE Transactions on* 11.3 (2002), p. 586–600. issn : 1045-9227 (cf. p. 155).

[WCR09] J. Wu, H. Christensen et J. Rehg. « Visual place categorization : Problem, dataset, and algorithm ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE. St. Louis, USA, 2009, p. 4763–4770 (cf. p. 159, 160, 184).

V. Détection visuelle des changements de lieux

Le chapitre IV nous a permis de présenter les bases de notre système de reconnaissance de lieux et d'étudier les paramètres qui influent le plus sur les performances. Nous avons souligné que jusqu'à présent, les transitions entre lieux étaient notifiées au système. Ce choix était justifié par le fait qu'il est possible de développer un mécanisme séparé de détection des seuils, par exemple en s'appuyant sur des capteurs lasers ([Pro+10]) [1].

Les travaux de Guillaume [Gui09] ont montré que, en l'absence d'informations sur les seuils, les performances sur la tâche de reconnaissance d'instances sont nettement moins bonnes. Ces travaux ont aussi montré que dans ce cas, il est plus intéressant d'utiliser une fenêtre d'intégration de taille plus petite.

Néanmoins, il est intéressant de développer un système de détection des seuils basé sur les informations visuelles afin de ne pas dépendre d'autres capteurs. Nous explorons quelques pistes dans ce chapitre.

1. Introduction

La problématique de la détection des transitions a été étudiée dans le SLAM topologique (voir [NP10]). Nous avons souligné que, dans ce cas, les lieux ne sont pas comparables à des lieux sémantiques. Une exception est constituée par les travaux de [RN99], dans lequel chaque nœud de la carte topologique est constitué par des pièces d'un environnement d'intérieur. Cependant, le système nécessite un apprentissage des transitions spécifiques à un environnement (le système n'est donc pas applicable à un nouvel environnement). Nous cherchons un système capable de fonctionner en ligne et qui ne nécessite pas un apprentissage spécifique.

Nous cherchons un moyen de détecter les transitions pour signaler au système tBoW quand re-initialiser la fenêtre d'intégration temporelle. Plus précisément, nous cherchons à donner au temps t la date du dernier changement de lieu $\delta(t)$. Un cas particulier de ce système consiste à classer chaque instant t comme étant ou non une transition. Le système est causal, donc $\delta(t) \leq t$.

1. Nous emploierons indifféremment les expressions « transition », « seuil » et « changement de lieux » pour désigner le passage d'un lieu à un autre.

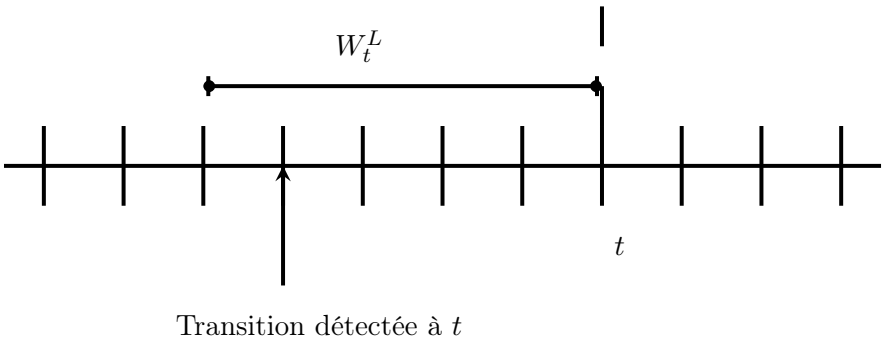


FIGURE V.1.: Adaptation de l'algorithme tBoW pour la détection en ligne de transitions.

Il faut légèrement adapter le système tBoW pour qu'il fonctionne avec la détection des seuils. En effet, l'intégration temporelle commence à partir de l'instant $\max\{\delta(t), t-L+1\}$. Si $t-L+1 < \delta(t) < t$ (voir Figure V.1), le système doit ré-initialiser la fenêtre et re-classifier les images comprises entre $\delta(t)$ et t.

Nous proposons d'utiliser les mêmes mécanismes pour la détection des transitions et pour la classification des lieux. La détection des transitions utilisera la même caractérisation des images et reposera sur une fenêtre de taille finie.

Cependant, l'intégration temporelle est indépendante de la méthode de détection des transitions. On peut donc utiliser des fenêtres de tailles différentes pour ces deux processus. Nous noterons ℓ la taille de la fenêtre utilisée pour détecter les transitions. De plus, W_t^ℓ désigne la fenêtre commençant à l'instant t et contenant les ℓ dernières observations, c'est-à-dire $\{x_{t-\ell+1}, \ldots, x_t\}$. Le symbole L désigne toujours la taille de la fenêtre utilisée pour l'intégration temporelle.

Nous présentons dans la section V.2 les études préliminaires qui nous ont permis de construire le système. La section V.3 présentera quelques possibilités pour détecter les seuils. Enfin, nous évaluons les performances de ces solutions pour la détection de changement de lieux et présentons les performances obtenues par tBoW lorsqu'on le couple à l'un de ces systèmes dans la section V.4.

2. Comportement de la classification au voisinage des seuils

Dans un premier temps, nous souhaitons étudier le comportement du système au voisinage des transitions lorsque le système n'est pas informé des changements de lieux.

Supposons que le robot passe du lieu c^1 au le lieu c^2 (voir Figure V.2). Dans le cas d'une fenêtre située loin de la transition (par exemple la fenêtre $W_{t_1}^L$ dans la Figure V.2), tous les

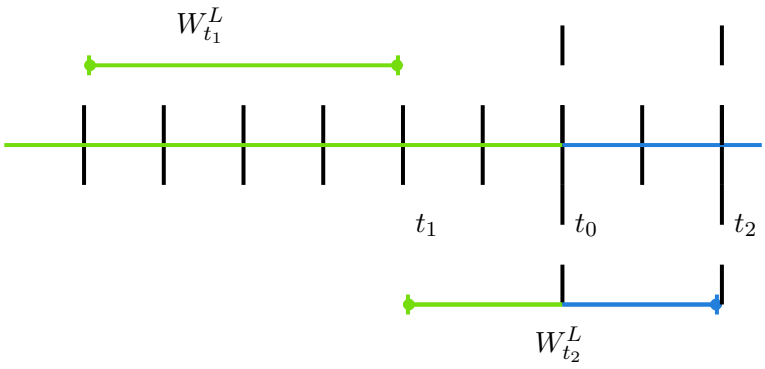


FIGURE V.2.: Comportement des fenêtres d'intégration temporelle lorsque l'on n'utilise pas la notification des changements de lieux : la transition réelle a lieu à t_0 ; les fenêtres au voisinage de la transitions contiennent des vues de différents lieux.

prototypes sont caractéristiques du lieu courant. Dans ce cas, l'intégration temporelle conduit rapidement à ce que la vraisemblance d'un lieu soit beaucoup plus grande que celle des autres lieux, ce qui se traduit par une distribution de probabilité *a posteriori* où une classe a une probabilité très proche de 1 et les autres une probabilité proche de 0. Le système est donc très sûr de sa réponse (qui est en général la bonne).

Au contraire, plaçons nous au temps t_2, situé quelques pas de temps après la transition. Dans ce cas, la fenêtre $W_{t_2}^L$ va contenir des images provenant de différents lieux. Les prototypes situés à $t' \leq t_0$ sont caractéristiques du lieu c^1, donc la vraisemblance $\mathrm{P}\left(x_{t'}|c^1\right)$ est grande. Au contraire, les prototypes trouvés à $t' > t_0$ sont caractéristiques du lieu c^2 donc la vraisemblance $\mathrm{P}\left(x_{t'}|c^2\right)$ est grande. On va donc obtenir une vraisemblance assez grande pour les deux lieux c^1 et c^2 : la distribution de probabilité *a posteriori* est plus étalée : le système « hésite » entre deux hypothèses assez probables.

La propriété qui permet de quantifier l'incertitude est l'entropie de la distribution, définie par :

$$H_t = -\sum_{y_t \in \mathcal{C}} \mathrm{P}\left(y_t|x_{t-\ell+1:t}\right) \times \log_2\left(\mathrm{P}\left(y_t|x_{t-\ell+1:t}\right)\right) \tag{V.1}$$

L'entropie est une grandeur caractéristique de la distribution de probabilité $\mathrm{P}\left(y_t|x_{t-L+1:t}\right)$ en bits. Elle est maximale quand la distribution de probabilité est uniforme, ce qui correspond à la situation d'incertitude maximale : toutes les classes ayant la même probabilité, le système n'a aucune raison de choisir un lieu plutôt qu'un autre. Dans le cas d'une distribution uniforme

avec p classes, l'entropie maximale est donnée par :

$$\begin{aligned} H_{max} &= -\sum_{y \in \mathcal{C}} \frac{1}{p} \times \log_2 \left(\frac{1}{p}\right) \\ &= -\log_2 \left(\frac{1}{p}\right) \\ &= \log_2 (p) \end{aligned} \tag{V.2}$$

À l'opposé, il est facile de voir que si la probabilité d'une classe tend vers 1 (la probabilité des autres classes tendant alors vers 0), ce qui correspond au cas où l'incertitude est la plus faible, la distribution *a posteriori* aura une entropie nulle. On peut donc supposer que lorsqu'une fenêtre d'intégration contient une transition, son entropie est plus élevée. La section suivante présente quelques expériences réalisées en vue de quantifier cette intuition.

2.1. Comportement moyen de l'entropie autour des transitions

Nous cherchons ici à comprendre le comportement de l'entropie autour des transitions pour savoir si on pourrait l'utiliser comme détecteur de transitions. L'évolution de la distribution de probabilité *a posteriori* et de l'entropie dépend de la taille de la fenêtre ℓ.

2.1.1. Fenêtres de taille croissante

Dans un premier temps, nous étudions à t fixé l'entropie sur des fenêtre de tailles croissante donc couvrant de plus en plus d'images dans le passé (voir Figure V.3). L'idée ici est que, au voisinage des transitions, les fenêtres de grande taille vont avoir une entropie plus importante que les fenêtres de petite taille car elles comportent plus d'images du lieu précédent, d'où :

$$\begin{aligned} H\left(W_t^{\ell_2}\right) &< H\left(W_t^{\ell_1}\right) \\ H\left(W_t^{\ell_3}\right) &> H\left(W_t^{\ell_2}\right) \end{aligned}$$

À l'inverse si il n'y a pas de transition, l'entropie diminue quand ℓ augmente :

$$H\left(W_t^{\ell_3}\right) < H\left(W_t^{\ell_2}\right) < H\left(W_t^{\ell_1}\right)$$

Pour étudier cela, nous avons réalisé le protocole de reconnaissance d'instances décrit dans la Section A.1.4 pour différentes tailles de fenêtres ℓ (voir Table V.1) sans notifier les changements de lieux. La taille de la SOM est fixée à $S = 20$. À des fins d'illustration, nous utilisons le descripteur CENTRIST. Nous nous intéressons au cas où les conditions de

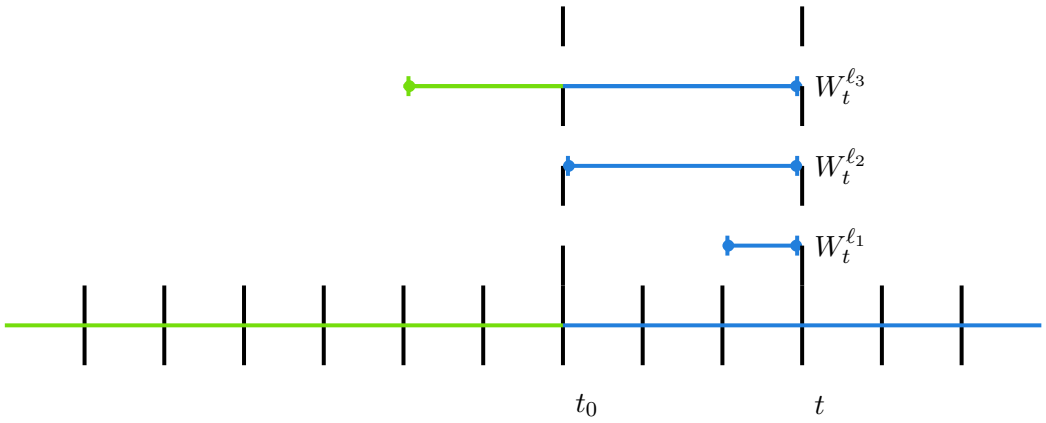


FIGURE V.3.: Utilisation de fenêtres de tailles différentes à l'instant t. La fenêtre de grande taille ℓ_3 inclut des images de plusieurs lieux, d'où une entropie plus forte.

1	3	5	7	10	15	20	30	50	70	100

TABLE V.1.: Valeur de ℓ utilisées.

luminosité en apprentissage et en reconnaissance sont les mêmes (conditions congruentes). Nous nous plaçons 20 images après la transition. La Figure V.4 donne l'entropie moyenne sur 1 467 transitions.

Si l'hypothèse était vérifiée, l'entropie à droite de la courbe en pointillés (qui marque la transition) devrait remonter. On n'observe pas cette remontée sur la figure.

L'interprétation probable est que la décroissance de l'entropie quand ℓ augmente l'emporte sur l'augmentation due à la transition.

2.1.2. Fenêtre de taille fixe

Dans un second temps, nous étudions l'évolution de l'entropie dans le temps autour des transitions (*i.e.* la dynamique de l'entropie), comme présenté dans la Figure V.5. L'idée ici est que, lors d'une transition, l'entropie passe par un maximum avant de décroître, d'où :

$$H\left(W_t^{\ell}\right) < H\left(W_{t-2}^{\ell}\right)$$
$$H\left(W_{t-4}^{\ell}\right) < H\left(W_{t-2}^{\ell}\right)$$

Alors qu'en dehors d'une transition, l'entropie est à peu près constante :

$$H\left(W_t^{\ell}\right) \approx H\left(W_{t-2}^{\ell}\right) \approx H\left(W_{t-4}^{\ell}\right)$$

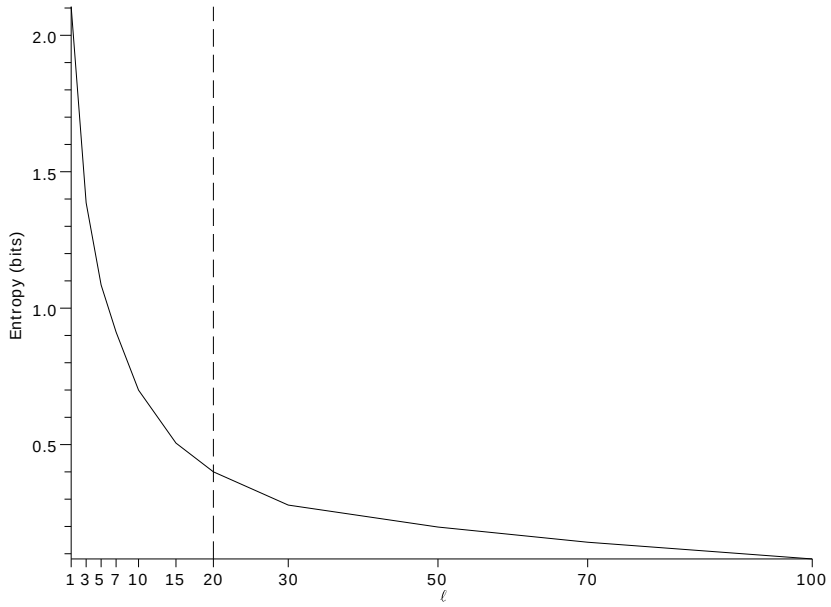


FIGURE V.4.: Entropie moyenne 20 pas de temps après la transition pour différentes valeurs de ℓ. La barre verticale montre la transition.

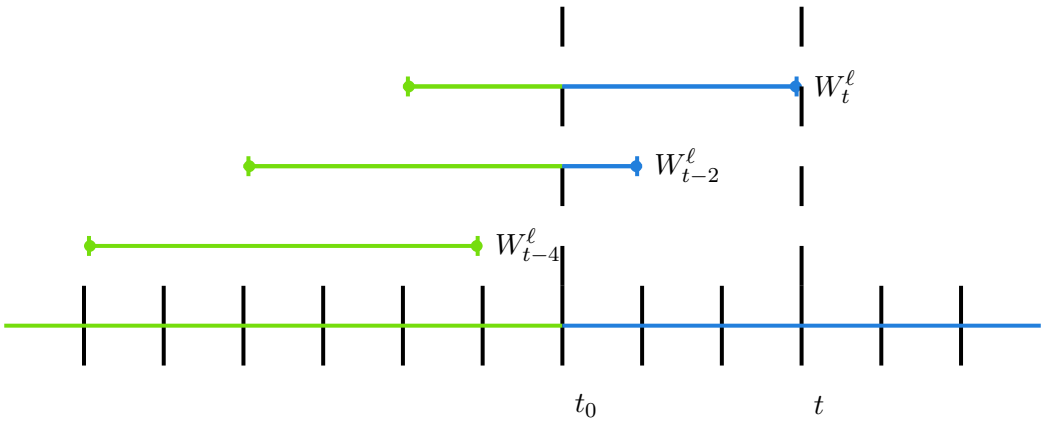


FIGURE V.5.: Utilisation d'une fenêtre de taille fixe pour la détection de transitions.

Pour cela, nous avons réalisé le même protocole que dans la section précédente. Comme dans la section précédente, nous nous intéressons aux conditions congruentes. Nous comparons l'entropie moyenne ainsi obtenue avec l'entropie moyenne sur des segments ne contenant pas de transition. L'entropie moyenne a été obtenue sur 1 468 transitions et 1 539 non-transitions.

La Figure V.6 montre les résultats pour le descripteur CENTRIST et $\ell = 30$. On peut observer plusieurs choses :

- premièrement, l'entropie est plus élevée à l'approche des transitions.
- deuxièmement, l'entropie présente un maximum à proximité d'une transition (centrée à $t = 0$), alors qu'elle est approximativement constante en-dehors.

Ces effets sont assez robustes : on peut les observer aussi pour les autres descripteurs et pour des fenêtres de grande taille. Cependant dans ce cas, la position du maximum intervient longtemps après la transition (étant donné qu'on prend en compte plus d'images du passé, il faut attendre plus longtemps pour que les images du nouveau lieu perturbent la classification).

Ces observations nous conduisent à nous intéresser à la détection des transitions à l'aide de fenêtres de petite taille. La section suivante présentent différentes méthodes inspirées par ces observations.

3. Présentation des algorithmes

3.1. Seuillage de l'entropie

L'idée la plus simple consiste à considérer que lorsque l'entropie dépasse un seuil donné H_θ, on a détecté une transition : Fixer un seuil est délicat mais ici, l'entropie est bornée donc on peut fixer le seuil en fonction de l'entropie maximale H_{max}.

Un autre problème vient du fait que l'entropie peut varier brutalement (voir Figure V.7). Afin de ne pas avoir trop de pics, nous utilisons la moyenne pondérée exponentiellement de l'entropie :

$$\tilde{H}_t = \alpha H_t + (1 - \alpha)\, \tilde{H}_{t-1} \tag{V.3}$$

Où α est une constante qui représente le taux de décroissance des poids et donc l'importance attribuée aux observations passées : plus α est grand moins on accorde d'importance aux observations passées. Notons que l'utilisation de ce filtrage ne remet pas en cause le fonctionnement en ligne du système.

La fonction δ_{seuil} est définie ainsi :

$$\delta_{seuil}(t) = \begin{cases} t & \text{si } \tilde{H}_t > H_\theta \quad \text{(V.4)} \\ \delta_{seuil}(t-1) & \text{sinon} \quad \text{(V.5)} \end{cases}$$

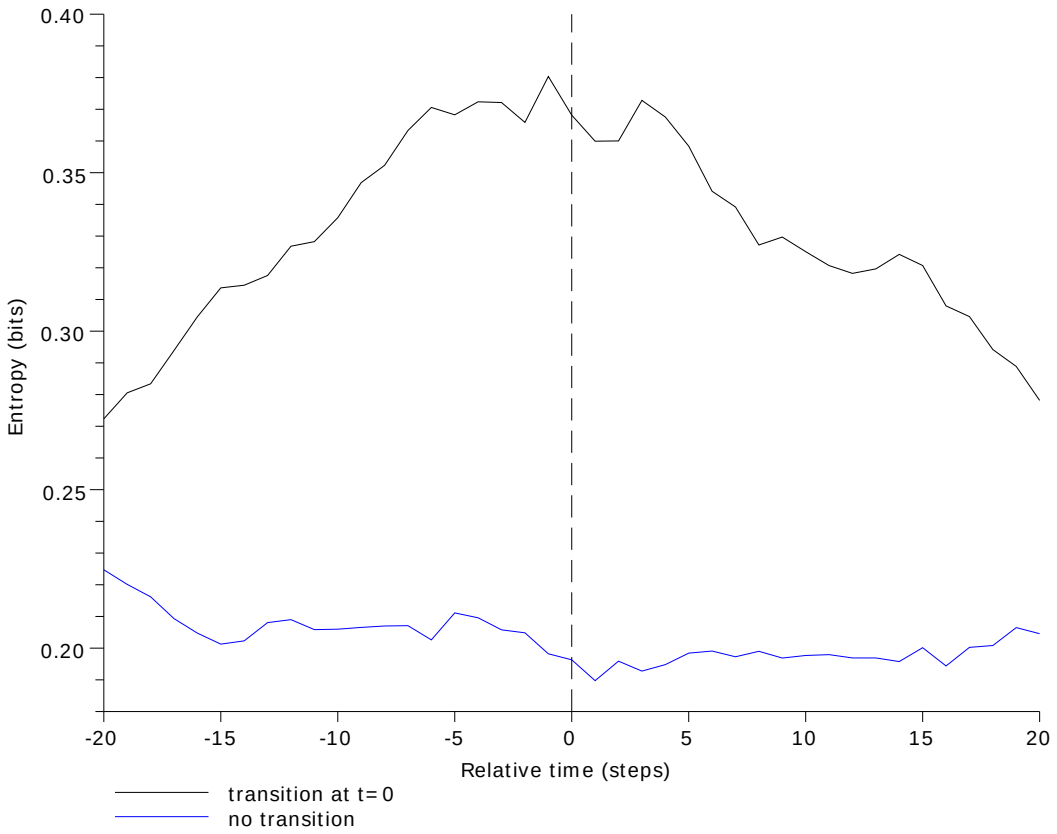


FIGURE V.6.: Évolution de l'entropie autour d'une transition à $t = 0$ et en dehors des transitions pour $\ell = 30$.

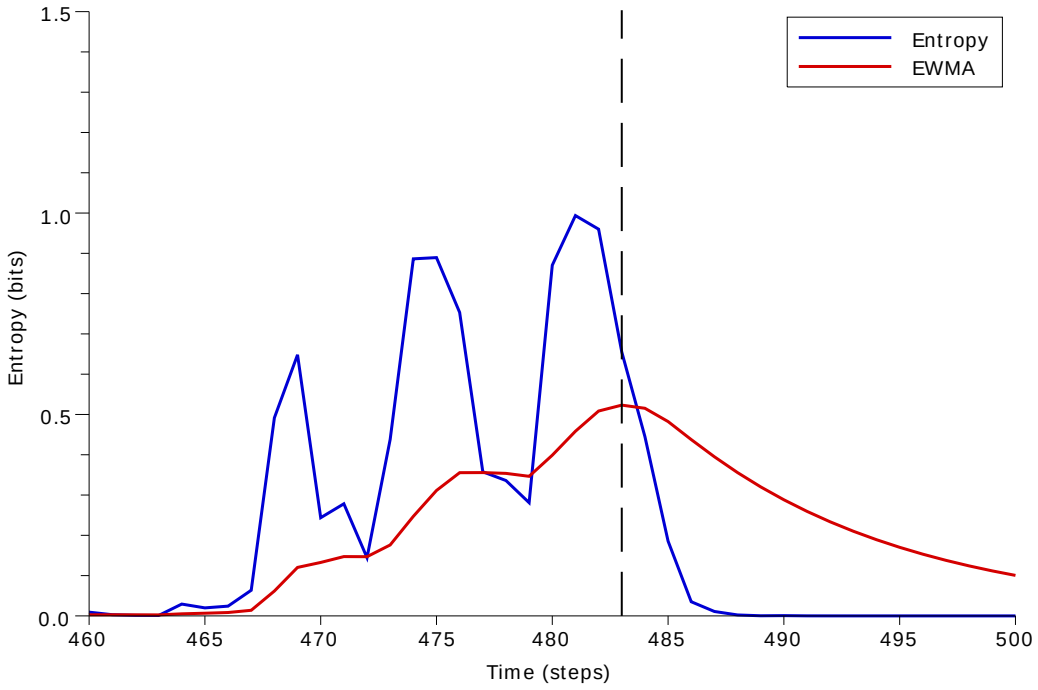
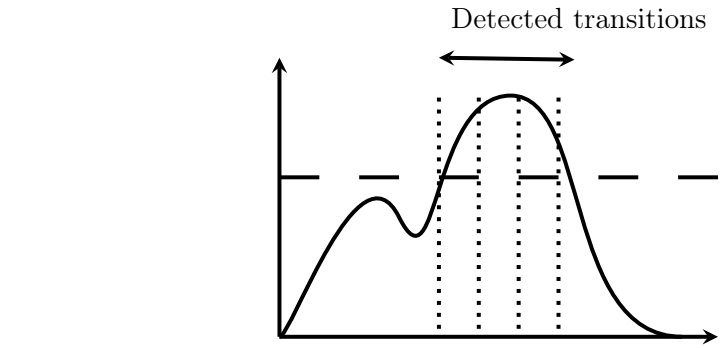
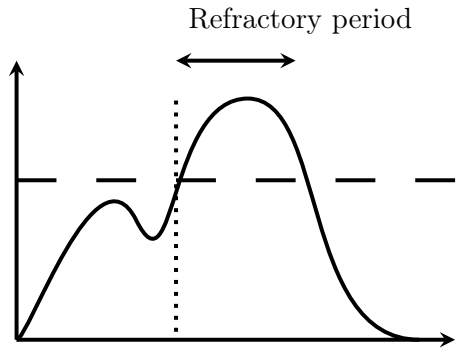


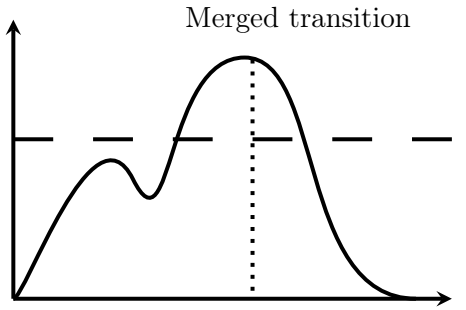
FIGURE V.7.: Lissage de l'entropie à l'aide d'une moyenne pondérée exponentiellement : la courbe bleue montre l'entropie non-lissée et la courbe rouge l'entropie lissée. Les données sont extraites d'une séquence réelle.



(a) Absence de filtrage des transitions.



(b) Utilisation d'une période réfractaire.



(c) Regroupement des transitions proches.

FIGURE V.8.: Filtrage des transitions détectées : (a) transitions détectées en l'absence de filtrage, (b) transition détectée en imposant une période réfractaire, (c) transition détectée en remplaçant un ensemble de transitions par leur médiane.

L'utilisation d'un seuil pose un autre problème : tous les points au-dessus du seuil vont déclencher une transition (voir figure V.8a). Pour éviter cela, nous avons donc besoin d'un mécanisme pour « filtrer » les transitions.

Le mécanisme le plus simple consiste à imposer une période réfractaire Δt_{refrac} entre deux transitions, ce qui correspond au fait que le robot reste un certain temps dans une pièce (voir figure V.8b).

Une autre manière de faire consiste à détecter des bouffées de transitions, c'est-à-dire un ensemble de transitions proches, et à les remplacer par une seule transition. Concrètement, on regroupe toutes les transitions séparées par moins de Δt_{burst} et on les remplace par la médiane (voir figure V.8c). Notons que dans ce cas, $\delta(t)$ change à chaque pas de temps.

3.2. Détection des maximas locaux de l'entropie

La Figure V.5 montre que l'entropie présente un maximum local au voisinage de la transition. On peut donc chercher à détecter les maxima locaux de l'entropie.

Comme pour le cas de la technique précédente, il est préférable de travailler avec l'entropie lissée, $\tilde{H}$. La détection se base sur le passage par 0 de la dérivée de $\tilde{H}$:

$$\delta_{max}(t) = \begin{cases} t & \text{si } \frac{\partial \tilde{H}_t}{\partial t} = 0 \quad \text{(V.6)} \\ \delta_{max}(t-1) & \text{sinon} \quad \text{(V.7)} \end{cases}$$

Il n'y a donc pas de paramètre supplémentaire (hormis α).

La Figure V.7 suggère qu'on va rencontrer plusieurs maxima locaux au voisinage de la transition donc à nouveau détecter des transitions proches. Nous proposons donc d'appliquer les mêmes techniques de filtrage pour éliminer les transitions trop proches.

3.3. Utilisation des changements de classification

La Figure V.6 montre que, en moyenne, l'entropie autour des transitions est plus élevée et présente un maximum local. Cependant, nous avons vu que l'entropie peut exhiber des variations brutales, ce qui rend difficile son utilisation dans un système réel.

La dernière idée consiste à utiliser les résultats de la classification pour des fenêtres de taille ℓ faible. Dans ce cas, une transition est détectée lorsque la classification change :

$$\delta_{classif}(t) = \begin{cases} t & \text{si } \overset{\star}{y}_{t-1} \neq \overset{\star}{y}_t \quad \text{(V.8)} \\ \delta_{classif}(t-1) & \text{sinon} \quad \text{(V.9)} \end{cases}$$

L'idée est que le résultat de la classification va rester plus stable que l'entropie : même si quelques images ambiguës viennent augmenter l'entropie, la classification ne change pas forcément. De plus, il n'y a pas de paramètre supplémentaire. Notons qu'on peut encore une fois utiliser les méthodes de filtrage afin d'enlever les détections trop proches.

Nous présentons dans la section suivante l'évaluation des différentes méthodes de détection et de filtrage.

4. Évaluation

À notre connaissance, seul Ranganathan [Ran10] publie des résultats sur la détection des changements de lieu. Ces résultats utilisent les séquences de la base VPC, sans apprentissage comme le lui permet son système. Notre système ne peut pas fonctionner dans ces conditions donc nous ne pouvons pas comparer directement les résultats.

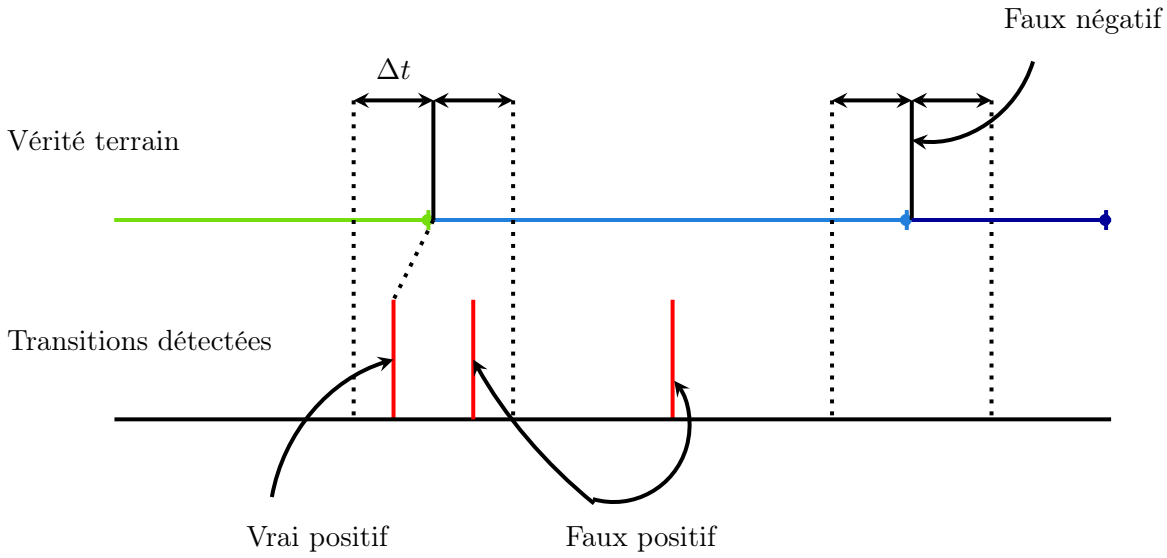


FIGURE V.9.: Évaluation de la détection des seuils : la première transition détectée est considérée comme correspondant à la première transition réelle ; la deuxième transition détectée est donc considérée comme un faux positif, de même que la troisième ; la deuxième transition réelle n'est pas détectée donc constitue un faux négatif.

Nous proposons donc d'utiliser les protocoles de reconnaissance d'instances et de catégorisation utilisés dans le chapitre IV. L'apprentissage est effectué de la même manière et avec les mêmes paramètres que dans le chapitre IV mais les changements de lieux ne sont pas notifiés dans la phase de reconnaissance.

L'évaluation procède en deux temps. Premièrement, nous évaluons la détection des transitions. Nous adoptons la même méthodologie d'évaluation que RANGANATHAN [Ran10], basée sur la théorie de la détection (voir Figure V.9 et Appendice D). Une transition détectée est considérée comme correcte si elle est à moins de $\Delta t = 10$ pas de temps d'une vraie transition, ce qui correspond à 2 s. Si une autre transition se trouve dans cet intervalle, elle est comptée comme un faux positif. De même, une transition détectée qui n'est située à proximité d'aucune transition réelle est un faux positif. Une transition réelle pour laquelle aucune transition n'est détectée dans un voisinage Δt est un faux négatif. Un vrai négatif correspondrait à l'absence de transition réelle et de transition détectée, ce qui n'est pas possible à évaluer ici (cela nécessiterait de classer tout les pas de temps, ce qui est incompatible avec la tolérance dans la détection des transitions).

La valeur de Δt a une grande influence sur l'évaluation des résultats. [Ran10] utilise $\Delta t = 20$ mais la la base VPC est échantillonnée à 10 Hz. La base COLD étant échantillonnée à 5 Hz,

Δt correspond à la même durée réelle. Nous avons pensé fixer Δt par rapport à la vitesse de déplacement du robot afin que Δt corresponde à la même distance métrique par rapport à la transition. Cependant, après discussion avec Jianxin Wu, il est assez difficile d'estimer la vitesse de déplacement du robot dans la base VPC. Nous avons donc décidé de conserver $\Delta t = 10$.

Deuxièmement, nous évaluons les résultats du système tBoW lorsqu'il est couplé avec un de ces systèmes de détection. Dans ce cas, nous utilisons une fenêtre de taille L pour l'intégration temporelle et nous comparons les résultats avec le cas où l'on ne notifie pas les transitions au système et avec le cas où les transitions réelles sont notifiées comme dans le chapitre IV.

Nous ne cherchons pas à évaluer complètement les différentes options mais juste à donner un aperçu des performances. Certains paramètres seront fixes :

- en nous appuyant sur le chapitre IV, nous n'utilisons que les signatures qui ont donné les meilleurs résultats sur la tâche étudiée.
- le coefficient de pondération α utilisé pour le lissage de l'entropie est fixé à $\alpha = 0{,}1$.
- pour la méthode de détection basée sur le seuillage de l'entropie, nous utilisons comme seuil $\frac{H_{max}}{2}$.
- pour la méthode de filtrage basée sur une période réfractaire, on fixe $\Delta t_{refrac} = 50$.
- pour la méthode de filtrage basée sur les bouffées de transitions, on fixe $\Delta t_{burst} = 20$.

Nous comparons les différentes approches de détection et de filtrage pour différentes valeurs de la fenêtre d'intégration $\ell \in \{10, 20, 30\}$.

4.1. Résultats en reconnaissance d'instances

4.1.1. Protocole

Nous utilisons le protocole de reconnaissance d'instances décrit dans la Section A.1.4. Nous nous intéressons aux conditions congruentes pour commencer (rappelons que la base VPC utilisée par [Ran10] a été acquise en essayant de réduire l'influence des conditions extérieures : volets baissés et lumières d'intérieur allumées). Dans ce cas, il y a 1 538 transitions réelles.

Conformément au chapitre IV, nous n'utilisons que les signatures qui ont donné les meilleurs résultats *i.e.* la signature CENTRIST et la signature GIST calculée sur les canaux de couleurs et projetée sur 80 dimensions. La taille de la SOM est fixée à $S = 20$. Pour l'intégration temporelle, nous utilisons $L = 200$. Avec ces paramètres :

- Si l'on n'utilise pas de mécanisme de détection des seuils, le taux de reconnaissance moyen est de 55,91% pour CENTRIST et de 48,90% pour GIST.
- Si l'on utilise la notification classique, le taux de reconnaissance moyen est de 78,63% pour CENTRIST et de 76,28% pour GIST.

4.1.2. Résultats & conclusion

Les résultats sont donnés dans la Table V.2 pour le descripteur CENTRIST et dans la Table V.3.

Globalement, toutes les options envisagées ici se caractérisent par un TFP assez élevé. Cependant, toutes les méthodes permettent d'augmenter nettement les performances sur la tâche de reconnaissance d'instances par rapport au cas où l'on ne notifie pas les changements de lieux. De ce point de vue, la méthode basée sur les changements de classification et la méthode basée sur les maxima locaux donnent les meilleurs résultats. Cependant, les résultats sont toujours nettement inférieurs au cas où l'on notifie les vrais changements de lieux.

Toujours d'un point de vue global, plus ℓ augmente, moins on détecte de transitions avant filtrage et plus les performances augmentent mais l'effet est assez faible.

Si l'on cherche à comparer les méthodes de détection, on peut faire les observations suivantes :

- la méthode basée sur le seuillage de l'entropie produit en général un nombre de détections très grand mais qui diminue fortement quand ℓ augmente. Ceci s'explique par le fait que l'entropie diminue quand ℓ augmente. Or nous utilisons toujours le même seuil d'où la diminution du nombre de détections. Le filtrage permet de réduire de façon importante le nombre de transitions détectées mais alors le TVP diminue fortement. De même, les performances de cette méthode sur la tâche de reconnaissance d'instances diminuent quand ℓ augmente (passant de 65,82% à 58,42% pour CENTRIST) et le TVP peut devenir très faible (inférieur à 1% pour GIST). Un meilleur réglage du seuil pourrait apporter des bénéfices mais en l'état, cette méthode ne semble pas efficace.
- la méthode basée sur les changements de classification et la méthode basée sur les maxima locaux donnent des résultats assez proches : la méthode basée sur les maxima a, en général, un TVP plus élevé mais le TFP est aussi plus élevé.

En ce qui concerne les méthodes de filtrage, la détection des bouffées donne les meilleures performances dans 6 cas sur 9 pour CENTRIST et dans 7 cas sur 9 pour GIST. Dans tout les cas, la performance obtenue avec cette méthode est proche de la meilleure performance. Le TVP est en général meilleur que la méthode basée sur la période réfractaire et le TFP est du même ordre.

Enfin, les deux descripteurs permettent d'atteindre des résultats assez proches avec un léger avantage pour CENTRIST. Le classification obtenue avec CENTRIST a une entropie plus élevée que celle avec le descripteur GIST, ce qui explique que la méthode basée sur le seuillage se dégrade moins brutalement dans le cas de CENTRIST.

ℓ	Détection	Filtrage	# détections	TVP	TFP	% correct
10	Seuillage entropie	Aucun	58 479	54,81 %	98,56 %	61,90
		Période réfractaire	2 425	25,36 %	83,92 %	65,43
		Détection bouffées	4 287	43,37 %	84,44 %	**65,82**
	Maxima locaux	Aucun	22 090	89,21 %	93,79 %	62,79
		Période réfractaire	4 378	33,75 %	88,15 %	**68,83***
		Détection bouffées	8 453	69,18 %	87,41 %	66,30
	Chgt. classif.	Aucun	32 856	85,44 %	96,00 %	62,05
		Période réfractaire	3 776	29,97 %	87,79 %	65,77
		Détection bouffées	7 063	65,02 %	85,84 %	**66,49**
20	Seuillage entropie	Aucun	13 614	15,99 %	98,19 %	60,02
		Période réfractaire	833	9,17 %	83,07 %	60,57
		Détection bouffées	1 199	12,03 %	84,57 %	**60,90**
	Maxima locaux	Aucun	16 564	81,73 %	92,41 %	65,48
		Période réfractaire	3 802	35,70 %	85,56 %	**69,86**
		Détection bouffées	6 808	65,21 %	85,27 %	68,03
	Chgt. classif.	Aucun	17 519	72,82 %	93,61 %	67,45
		Période réfractaire	2 975	36,74 %	81,01 %	68,79
		Détection bouffées	4 905	58,00 %	81,81 %	**69,96***
30	Seuillage entropie	Aucun	4 670	5,66 %	98,14 %	58,12
		Période réfractaire	356	3,58 %	84,55 %	58,26
		Détection bouffées	470	4,16 %	86,38 %	**58,42**
	Maxima locaux	Aucun	13 310	77,63 %	91,03 %	67,59
		Période réfractaire	3 391	38,49 %	82,54 %	**70,46**
		Détection bouffées	5 804	62,94 %	83,32 %	69,32
	Chgt. classif.	Aucun	11 738	64,50 %	91,55 %	70,20
		Période réfractaire	2 519	38,10 %	76,74 %	70,91
		Détection bouffées	3 864	54,81 %	78,18 %	**71,50***

TABLE V.2.: Résultats de la détection des seuils pour le protocole de reconnaissance d'instances avec le descripteur CENTRIST. Le nombre en gras est la meilleure performance pour un ℓ et une méthode de détection donnés. La meilleure performance pour un ℓ donné est marquée d'une astérisque.

ℓ	Détection	Filtrage	# détections	TVP	TFP	% correct
10	Seuillage entropie	Aucun	8 383	6,37 %	98,83 %	55,38
		Période réfractaire	536	3,84 %	88,99 %	54,35
		Détection bouffées	745	4,62 %	90,47 %	**55,66**
	Maxima locaux	Aucun	11 231	74,32 %	89,82 %	68,60
		Période réfractaire	3 682	31,34 %	86,91 %	68,89
		Détection bouffées	6 213	60,92 %	84,92 %	**69,36***
	Chgt. classif.	Aucun	13 181	75,81 %	91,15 %	65,97
		Période réfractaire	3 086	37,52 %	81,30 %	66,04
		Détection bouffées	5 110	61,25 %	81,57 %	**68,19**
20	Seuillage entropie	Aucun	1 958	1,50 %	98,83 %	**51,61**
		Période réfractaire	143	0,65 %	93,01 %	51,16
		Détection bouffées	189	1,04 %	91,53 %	51,59
	Maxima locaux	Aucun	7 596	66,64 %	86,51 %	69,52
		Période réfractaire	3 019	37,19 %	81,05 %	69,33
		Détection bouffées	4 690	56,96 %	81,32 %	**70,43***
	Chgt. classif.	Aucun	7 549	63,72 %	87,02 %	68,71
		Période réfractaire	2 549	40,12 %	75,79 %	68,60
		Détection bouffées	3 861	53,90 %	78,53 %	**70,15**
30	Seuillage entropie	Aucun	567	0,65 %	98,24 %	**50,11**
		Période réfractaire	40	0,46 %	82,50 %	49,90
		Détection bouffées	56	0,65 %	82,14 %	50,08
	Maxima locaux	Aucun	5 814	55,14 %	85,41 %	70,15
		Période réfractaire	2 615	35,44 %	79,16 %	70,21
		Détection bouffées	3 814	47,85 %	80,70 %	**70,55**
	Chgt. classif.	Aucun	5 385	49,93 %	85,74 %	69,95
		Période réfractaire	2 257	34,07 %	76,78 %	69,80
		Détection bouffées	3 158	44,67 %	78,25 %	**70,60***

TABLE V.3.: Résultats de la détection des seuils pour le protocole de reconnaissance d'instances avec le descripteur GIST calculé sur les canaux de couleurs et projeté sur 80 dimensions. Le nombre en gras est la meilleure performance pour un ℓ et une méthode de détection donnés. La meilleure performance pour un ℓ donné est marquée d'une astérisque.

4.2. Résultat en catégorisation

4.2.1. Protocoles

Conformément au chapitre IV, nous n'utilisons que les signatures qui ont donné les meilleurs résultats *i.e.* la signature CENTRIST et la signature GIST calculée sur les canal de luminance et projetée sur 80 dimensions. La taille de la SOM est fixée à $S = 35$. Pour l'intégration temporelle, nous utilisons $L = 60$.

Nous utilisons le protocole de catégorisation sur la base COLD décrit dans la Section A.1.5. Il y a 405 transitions. Avec $L = 60$:

- Si l'on n'utilise pas de mécanisme de détection des seuils, le TVP moyen est de 33,39% pour CENTRIST et de 36,70% pour GIST.
- Si l'on utilise la notification classique, le TVP est de 32,40% pour CENTRIST et de 35,89% pour GIST.

On voit dans ce cas que la notification des changements de lieux diminue légèrement les résultats.

Nous utilisons aussi le protocole sur la base VPC décrit dans la Section A.2.3. Le nombre de transitions est 76. Avec $L = 60$:

- Si l'on n'utilise pas de mécanisme de détection des seuils, le TVP moyen est de 33,22% pour CENTRIST et de 32,69% pour GIST.
- Si l'on utilise la notification classique, le TVP est de 33,92% pour CENTRIST et de 32,89% pour GIST.

4.2.2. Résultats & conclusion

Concernant le protocole sur la base COLD, les résultats sont donnés dans la Table V.4 pour le descripteur CENTRIST et dans la Table V.5 pour GIST.

Concernant le protocole sur la base VPC, les résultats sont donnés dans la Table V.6 pour le descripteur CENTRIST et dans la Table V.7 pour GIST.

La notification des changements réels de lieu améliore peu (voire dégrade) les performances moyennes du système. Contrairement à ce qu'on observe sur le protocole de reconnaissance d'instances, il existe des cas où les performances sont meilleures en utilisant un système de détection des transitions qu'en notifiant les vrais changements de lieux au système. Par exemple, sur la base COLD, avec le descripteur GIST, le système basé sur la détection des maxima locaux avec $\ell = 10$, couplé au filtrage par période réfractaire (voir Table V.5), le TVP est supérieur à 38%, alors qu'il est de 35,89% lorsqu'on notifie les vrais changements de lieux.

Notons que dans ce cas, le TFP est supérieur à 96% et qu'on détecte près de 2 600 transitions, ce qui signifie que la fenêtre utilisée pour la classification est souvent re-initialisée.

ℓ	Détection	Filtrage	# détections	TVP	TFP	% TVP moyen
10	Seuillage entropie	Aucun	182 725	98,27 %	99,78 %	28,96
		Période réfractaire	4 006	42,72 %	95,68 %	30,37
		Détection bouffées	9 408	89,14 %	96,16 %	**30,84**
	Maxima locaux	Aucun	16 163	84,94 %	97,87 %	30,71
		Période réfractaire	3 345	38,52 %	95,34 %	31,04
		Détection bouffées	6 456	65,68 %	95,88 %	**31,22***
	Chgt. classif.	Aucun	25 425	92,59 %	98,53 %	30,21
		Période réfractaire	3 327	44,20 %	94,62 %	30,11
		Détection bouffées	6 565	69,38 %	95,72 %	**30,71**
20	Seuillage entropie	Aucun	122 606	81,73 %	99,73 %	30,35
		Période réfractaire	3 371	39,26 %	95,28 %	31,30
		Détection bouffées	7 049	68,15 %	96,08 %	**31,95***
	Maxima locaux	Aucun	12 831	81,48 %	97,43 %	31,12
		Période réfractaire	3 151	34,81 %	95,53 %	30,78
		Détection bouffées	5 869	61,48 %	95,76 %	**31,55**
	Chgt. classif.	Aucun	15 909	78,27 %	98,01 %	31,23
		Période réfractaire	2 961	33,33 %	95,44 %	30,52
		Détection bouffées	5 312	59,26 %	95,48 %	**31,49**
30	Seuillage entropie	Aucun	81 552	58,77 %	99,71 %	31,56
		Période réfractaire	2 549	27,41 %	95,65 %	31,73
		Détection bouffées	4 977	47,65 %	96,12 %	**32,57***
	Maxima locaux	Aucun	11 685	81,23 %	97,18 %	31,65
		Période réfractaire	3 082	34,81 %	95,43 %	31,55
		Détection bouffées	5 593	63,21 %	95,42 %	**31,82**
	Chgt. classif.	Aucun	11 980	61,98 %	97,90 %	**31,47**
		Période réfractaire	2 706	29,88 %	95,53 %	31,23
		Détection bouffées	4 503	48,89 %	95,60 %	31,39

TABLE V.4.: Résultats de la détection des seuils pour le protocole de catégorisation sur la base COLD avec le descripteur CENTRIST. Le nombre en gras est la meilleure performance pour un ℓ et une méthode de détection donnés. La meilleure performance pour un ℓ donné est marquée d'une astérisque.

ℓ	Détection	Filtrage	# détections	TVP	TFP	% TVP moyen
10	Seuillage entropie	Aucun	164 674	88,64 %	99,78 %	31,89
		Période réfractaire	3 837	37,78 %	96,01 %	**35,56**
		Détection bouffées	8 758	76,30 %	96,47 %	34,38
	Maxima locaux	Aucun	6 264	44,69 %	97,11 %	36,27
		Période réfractaire	2 599	19,75 %	96,92 %	**38,34***
		Détection bouffées	4 142	34,57 %	96,62 %	36,54
	Chgt. classif.	Aucun	9 834	51,60 %	97,87 %	32,09
		Période réfractaire	2 471	19,75 %	96,76 %	**37,82**
		Détection bouffées	4 219	40,74 %	96,09 %	33,23
20	Seuillage entropie	Aucun	134 468	74,07 %	99,78 %	33,96
		Période réfractaire	3 426	36,30 %	95,71 %	**36,39**
		Détection bouffées	7 327	63,95 %	96,47 %	36,07
	Maxima locaux	Aucun	4 917	39,01 %	96,79 %	35,94
		Période réfractaire	2 349	19,51 %	96,64 %	**38,33***
		Détection bouffées	3 641	32,59 %	96,37 %	36,42
	Chgt. classif.	Aucun	6 736	53,33 %	96,79 %	34,00
		Période réfractaire	2 221	15,06 %	97,25 %	**36,94**
		Détection bouffées	3 796	39,51 %	95,79 %	34,63
30	Seuillage entropie	Aucun	93 490	52,59 %	99,77 %	34,95
		Période réfractaire	2 557	26,67 %	95,78 %	**36,87**
		Détection bouffées	5 225	44,69 %	96,54 %	35,84
	Maxima locaux	Aucun	4 329	40,99 %	96,17 %	36,16
		Période réfractaire	2 209	20,74 %	96,20 %	**37,64***
		Détection bouffées	3 346	37,28 %	95,49 %	36,30
	Chgt. classif.	Aucun	5 187	43,21 %	96,63 %	35,30
		Période réfractaire	2 033	19,75 %	96,06 %	**36,85**
		Détection bouffées	3 330	37,78 %	95,41 %	35,61

TABLE V.5.: Résultats de la détection des seuils pour le protocole de catégorisation sur la base COLD avec le descripteur GIST calculé sur le canal de luminance et projeté sur 80 dimensions. Le nombre en gras est la meilleure performance pour un ℓ et une méthode de détection donnés. La meilleure performance pour un ℓ donné est marquée d'une astérisque.

ℓ	Détection	Filtrage	# détections	TVP	TFP	% TVP moyen
10	Seuillage entropie	Aucun	6 313	45,45 %	99,60 %	33,70
		Période réfractaire	262	14,55 %	96,95 %	**34,09**
		Détection bouffées	461	36,36 %	95,66 %	33,50
	Maxima locaux	Aucun	2 488	92,73 %	97,95 %	32,69
		Période réfractaire	447	32,73 %	95,97 %	**33,51**
		Détection bouffées	895	65,45 %	95,98 %	32,72
	Chgt. classif.	Aucun	3 707	98,18 %	98,54 %	32,39
		Période réfractaire	415	52,73 %	93,01 %	**34,26***
		Détection bouffées	820	65,45 %	95,61 %	33,81
20	Seuillage entropie	Aucun	1 502	12,73 %	99,53 %	**33,22**
		Période réfractaire	87	7,27 %	95,40 %	32,94
		Détection bouffées	123	9,09 %	95,93 %	33,15
	Maxima locaux	Aucun	2 035	83,64 %	97,74 %	31,75
		Période réfractaire	410	41,82 %	94,39 %	**32,88**
		Détection bouffées	765	70,91 %	94,90 %	32,77
	Chgt. classif.	Aucun	2 194	74,55 %	98,13 %	33,31
		Période réfractaire	350	40,00 %	93,71 %	33,13
		Détection bouffées	629	58,18 %	94,91 %	**34,08***
30	Seuillage entropie	Aucun	594	1,82 %	99,83 %	**33,70***
		Période réfractaire	41	0,00 %	100,00 %	33,31
		Détection bouffées	55	0,00 %	100,00 %	33,44
	Maxima locaux	Aucun	1 648	67,27 %	97,75 %	33,39
		Période réfractaire	379	30,91 %	95,51 %	33,04
		Détection bouffées	665	52,73 %	95,64 %	**33,52**
	Chgt. classif.	Aucun	1 566	58,18 %	97,96 %	33,45
		Période réfractaire	310	23,64 %	95,81 %	33,61
		Détection bouffées	515	45,45 %	95,15 %	**33,68**

TABLE V.6.: Résultats de la détection des seuils pour le protocole de catégorisation sur la base VPC avec le descripteur CENTRIST. Le nombre en gras est la meilleure performance pour un ℓ et une méthode de détection donnés. La meilleure performance pour un ℓ donné est marquée d'une astérisque.

ℓ	Détection	Filtrage	# détections	TVP	TFP	% TVP moyen
10	Seuillage entropie	Aucun	1 768	7,27 %	99,77 %	32,36
		Période réfractaire	82	0,00 %	100,00 %	32,35
		Détection bouffées	131	1,82 %	99,24 %	**32,38***
	Maxima locaux	Aucun	1 816	96,36 %	97,08 %	28,62
		Période réfractaire	432	41,82 %	94,68 %	**31,73**
		Détection bouffées	819	78,18 %	94,75 %	29,05
	Chgt. classif.	Aucun	2 395	94,55 %	97,83 %	28,47
		Période réfractaire	420	38,18 %	95,00 %	**30,73**
		Détection bouffées	802	74,55 %	94,89 %	30,13
20	Seuillage entropie	Aucun	447	3,64 %	99,55 %	32,55
		Période réfractaire	25	1,82 %	96,00 %	**32,78***
		Détection bouffées	38	3,64 %	94,74 %	32,68
	Maxima locaux	Aucun	1 427	81,82 %	96,85 %	29,13
		Période réfractaire	398	38,18 %	94,72 %	29,72
		Détection bouffées	722	63,64 %	95,15 %	**30,48**
	Chgt. classif.	Aucun	1 535	78,18 %	97,20 %	29,88
		Période réfractaire	381	29,09 %	95,80 %	**31,22**
		Détection bouffées	687	60,00 %	95,20 %	31,13
30	Seuillage entropie	Aucun	124	1,82 %	99,19 %	32,76
		Période réfractaire	11	1,82 %	90,91 %	32,73
		Détection bouffées	12	1,82 %	91,67 %	**32,76***
	Maxima locaux	Aucun	1 205	69,09 %	96,85 %	29,54
		Période réfractaire	379	30,91 %	95,51 %	30,22
		Détection bouffées	653	56,36 %	95,25 %	**31,26**
	Chgt. classif.	Aucun	1 252	63,64 %	97,20 %	29,85
		Période réfractaire	361	27,27 %	95,84 %	30,82
		Détection bouffées	606	45,45 %	95,87 %	**30,84**

TABLE V.7.: Résultats de la détection des seuils pour le protocole de catégorisation sur la base VPC avec le descripteur GIST calculé sur le canal de luminance et projeté sur 80 dimensions. Le nombre en gras est la meilleure performance pour un ℓ et une méthode de détection donnés. La meilleure performance pour un ℓ donné est marquée d'une astérisque.

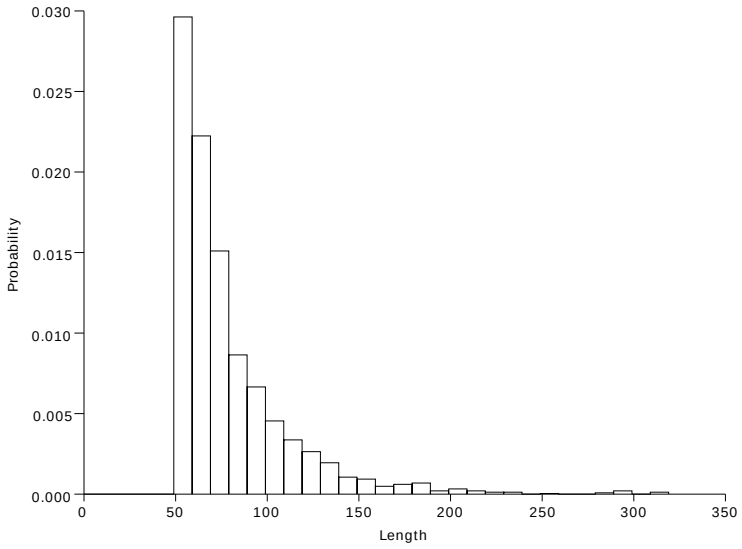


FIGURE V.10.: Distribution de l'écart entre deux transitions détectées pour le descripteur GIST : les transitions sont détectées lors des maxima locaux avec $\ell = 10$ et on impose une période réfractaire $\Delta t_{refrac} = 50$ (ce qui explique qu'on n'a pas de valeurs en dessous de 50 pas de temps.

La Figure V.10 montre la distribution de l'écart entre deux transitions détectées. La longueur moyenne entre deux transitions détectées est $\bar{L} \approx 79$ (la médiane est $\widehat{L} = 69$). En tenant compte de la re-initialisation de la fenêtre, en moyenne, la fenêtre utilisée a une taille $\overset{\star}{L} \approx 35$.

Nous avons donc comparé les résultats avec l'utilisation d'une fenêtre de taille $L = 35$ mais sans détection des transitions. Dans ce cas, le TVP moyen est de 35,80% (de même, pour $L = 40$ et $L = 50$, les résultats sont inférieurs). Il y a donc bien un effet de la détection des transitions.

Cela peut s'expliquer par le fait que notre système détecte les moments où la classification devient confuse. À ces moments là, il est plus avantageux de ne pas utiliser de mécanisme d'intégration temporelle qui peut piéger le système dans une hypothèse fausse.

Quoi qu'il en soit, les résultats en catégorisation sont limités par la robustesse de notre méthode aux variations environnementales (voir Section IV.5.3).

5. Conclusion

Nous avons présenté dans ce chapitre des travaux sur la détection de seuils. Les méthodes que nous avons développées s'appuient sur l'incertitude de la classification obtenue en utilisant une fenêtre d'intégration courte. Ces méthodes peuvent fonctionner en-ligne et ne nécessitent pas d'apprentissage supplémentaire.

Les résultats sur la tâche de reconnaissance d'instances, dans laquelle nous obtenons de bons résultats, sont intéressants. Les meilleurs systèmes permettent d'augmenter le taux de reconnaissance de plus de 20% pour le descripteur GIST.

En revanche, comme nous utilisons les résultats de la classification, si celle-ci n'est pas bonne, la détection des seuils est nettement moins efficace, ce qui est le cas dans la tâche de catégorisation. Cependant, nous avons montré que, bien que le TVP soit en général assez faible, les mécanismes employés peuvent conduire à une amélioration de la classification en repérant les moments où il est préférable de ne pas utiliser l'intégration temporelle.

Plusieurs moyens sont envisageables pour améliorer la détection des seuils en utilisant l'entropie. Tout d'abord, on pourrait mieux étudier les paramètres afin de diminuer le TFP. De plus, il serait intéressant d'étudier en détail les endroits où le système détecte des transitions pour faire le lien avec les caractéristiques de ces images. On pourrait aussi envisager d'apprendre les seuils comme dans [RN99]. Une dernière extension envisageable serait de reconnaître le profil temporel de l'entropie avant une transition. Enfin, toute amélioration des modèles de lieux conduit à l'amélioration de la détection des transitions.

Bibliographie

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 189).

[NP10] N. Nourani-Vatani et C. Pradalier. « Scene change detection for vision-based topological mapping and localization ». Dans : *IROS*. IEEE, 2010, p. 3792–3797. isbn : 978-1-4244-6674-0 (cf. p. 189).

[Pro+10] A. Pronobis et al. « Multi-modal semantic place classification ». Dans : *The International Journal of Robotics Research* 29.2-3 (2010), p. 298–320 (cf. p. 189).

[Ran10] A. Ranganathan. « PLISS : Detecting and Labeling Places using Online Change-point Detection ». Dans : *Proceedings of the 2010 Robotics : Science and Systems Conference (RSS 2010)*. 2010 (cf. p. 199–201).

[RN99] D. Radhakrishnan et I. Nourbakhsh. « Topological robot localization by training a vision-based transition detector ». Dans : *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*. T. 1. IEEE. 1999, p. 468–473 (cf. p. 189, 211).

VI. BFMC : Utilisation de l'ordre local pour la reconnaissance de lieux

L'hypothèse fondamentale du système tBoW présenté au chapitre IV est que l'ordre d'apparition des mots n'importe pas. Dans ce chapitre, nous présentons une méthode de reconnaissance de lieux qui vise à dépasser cette hypothèse. Une partie des travaux a été suggéré par le système présenté dans [Gui09, chap. 8].

1. Introduction

L'approche *temporal Bag-of-Words (tBoW)* présentée au chapitre IV ne tient pas compte de l'ordre des mots. Cela est justifié d'une part par l'utilisation d'une fenêtre finie et d'autre par le fait que l'ordre d'exploration des pièces peut changer. Il n'est donc pas nécessaire de modéliser complètement l'ordre d'apparition des mots.

Cependant, les capacités sensori-motrices du robot sont limitées : une fois le robot entré dans une pièce, ses déplacements sont contraints par la configuration de celle-ci. Les actions possibles sont donc limitées, ce qui va se traduire par une suite d'entrées corrélées. On peut donc dire que localement l'ordre des observations n'est pas quelconque. On peut dès lors se poser la question d'exploiter cet ordre local sans modéliser complètement l'ordre d'apparition des séquences.

Les modèles basés sur le filtrage bayésien (voir Section I.1.4 et Section III.4.4) permettent de prendre en compte la corrélation entre les observations à différents instants. En effet, dans un tel modèle, les observations sont générées par un état caché qui lui même évolue en fonction du temps. Cependant, ce modèle repose de manière cruciale sur l'hypothèse de l'état complet (voir Section I.1.4) qui stipule que la connaissance de l'état courant (un lieu sémantique dans notre cas) explique complètement l'observation. Si l'on connaît l'état caché, les différentes observations sont indépendantes. Il n'y a donc pas de prise en compte de l'ordre local. Comme nous l'avons souligné, il y a un gouffre sémantique important entre la notion de lieu sémantique et les informations bas-niveau obtenues à partir de l'image.

Nous avons donc proposé un modèle fondé sur les mêmes bases que le modèle tBoW mais prenant en compte l'ordre local. Dans ce modèle, l'état caché explique les observations mais

celles-ci dépendent aussi des observations précédentes. L'idée générale est que la connaissance du lieu seul (l'état caché) ne permet pas d'expliquer complètement les observations (les images).

Nous présentons dans la section VI.2 le modèle que nous avons développé. Les sections VI.3 et VI.4 seront consacrées à l'évaluation du système sur les tâche de reconnaissance d'instances et de catégorisation respectivement. De plus, nous comparerons notre algorithme avec l'approche classique en filtrage bayésien décrite Section III.4.4.1.

2. Filtrage bayésien sur chaînes de Markov

2.1. Présentation générale du modèle

Le modèle que nous proposons est basé lui aussi sur la notion de mots visuels. En particulier les signatures et les cartes de Kohonen utilisées sont les mêmes (voir Section IV.3.3 et Section IV.3.2).

Cependant, la manière d'intégrer les informations diffère. Comme nous l'avons dit, tBoW ne modélise pas l'ordre des mots et les transitions entre pièces. Le modèle utilisé pour tenir compte de l'ordre local est représenté de manière graphique dans la Figure VI.1 à comparer avec la Figure I.6 page 18. Comme dans la section précédente, nous ne prenons pas en compte les actions du robot. Comme dans le cas du filtrage bayésien classique, l'état évolue selon un processus markovien d'ordre 1. L'état courant dépend uniquement de l'état précédent *i.e.* :

$$\mathrm{p}\left(\boldsymbol{Y}_t|\boldsymbol{Y}_{0:t-1}, \boldsymbol{X}_{1:t-1}\right) = \mathrm{p}\left(\boldsymbol{Y}_t|\boldsymbol{Y}_{t-1}\right) \tag{VI.1}$$

En revanche, l'observation courante ne dépend pas uniquement de l'état courant mais aussi de l'observation passée :

$$\mathrm{p}\left(\boldsymbol{X}_t|\boldsymbol{Y}_{0:t}, \boldsymbol{X}_{1:t-1}\right) = \mathrm{p}\left(\boldsymbol{X}_t|\boldsymbol{Y}_t, \boldsymbol{X}_{t-1}\right) \tag{VI.2}$$

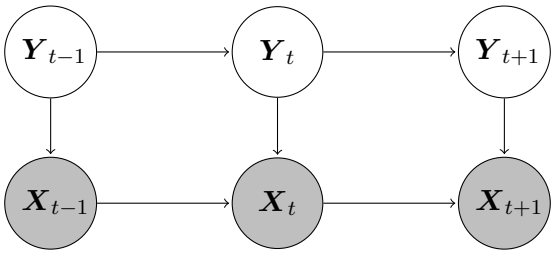


FIGURE VI.1.: Réseau bayésien décrivant notre modèle. Les nœuds clairs sont les variables cachées (inconnues).

Cette dépendance explicite des observations entre elles nous permet de modéliser l'ordre local. L'algorithme a été baptisé *Bayesian Filtering with Markov Chains (BFMC)*. Bien que nous l'ayons développé indépendamment, un tel modèle a déjà été étudié dans la littérature sous le nom de Modèle de Markov à double chaîne (*Double Chain Markov Model*,voir [Ber99]) ou de modèle de Markov auto-régressif (*Autoregressive Hidden Markov Model*, voir [ER05]). Ce modèle, par sa capacité à prendre en compte les corrélations entre les observations, a été utilisé notamment en reconnaissance de la parole (voir [ER05]) et pour des prévisions météorologiques (voir [Tan04 ; BBR10]).

2.2. Algorithme de filtrage bayésien

Nous présentons ici la règle de filtrage bayésien dans le cas de variables discrètes. En effet, l'état caché est le lieu sémantique (ou la catégorie) y_t et l'observation est donné par le prototype x_t, qui sont des variables discrètes. Les équations de base peuvent être écrites dans le cas de variables continues mais leur estimation devient beaucoup plus compliquée.

Le but de l'algorithme est de connaître l'état caché courant y_t étant donné la séquence des observations $x_{1:t}$:

$$\text{bel}\left(Y_t = c^i\right) = \text{P}\left(Y_t = c^i | X_{1:t}\right) \tag{VI.3}$$

La règle de Bayes nous donne :

$$\text{bel}\left(Y_t = c^i\right) = \frac{\text{P}\left(X_t | Y_t = c^i, X_{1:t-1}\right) \text{P}\left(Y_t = c^i | X_{1:t-1}\right)}{\eta} \tag{VI.4}$$

Où $\eta = \text{P}\left(X_t | X_{1:t-1}\right)$ est un terme de normalisation qui ne dépend pas de Y_t.

Par hypothèse :

$$\text{P}\left(X_t | Y_t = c^i, X_{1:t-1}\right) = \text{P}\left(X_t | Y_t = c^i, X_{t-1}\right) \tag{VI.5}$$

De plus, la formule des probabilités totales nous dit que :

$$\text{P}\left(Y_t = c^i | X_{1:t-1}\right) = \sum_{c^j} \text{P}\left(Y_t = c^i | Y_{t-1} = c^j, X_{1:t-1}\right) \text{P}\left(Y_{t-1} = c^j | X_{1:t-1}\right)$$

Par hypothèse, Y_t est complètement déterminé par Y_{t-1}, d'où :

$$\text{P}\left(Y_t = c^i | Y_{t-1} = c^j, X_{1:t-1}\right) = \text{P}\left(Y_t = c^i | Y_{t-1} = c^j\right) \tag{VI.6}$$

De plus, par définition, la distribution $\text{P}\left(Y_{t-1} | X_{1:t-1}\right)$ est $\text{bel}\left(Y_{t-1}\right)$.

En combinant VI.6 et VI.5 dans VI.4, on obtient :

$$\text{bel}\left(Y_t = c^i\right) \propto \text{P}\left(X_t | Y_t = c^i, X_{t-1}\right) \sum_{c^j} \text{P}\left(Y_t = c^i | Y_{t-1} = c^j\right) \text{bel}\left(Y_{t-1} = c^j\right) \tag{VI.7}$$

Le terme de normalisation η est obtenu en considérant que $\sum_{c^i} \text{bel}\,(Y_t = c^i) = 1$.

L'approche classique en filtrage bayésien est un cas particulier de l'algorithme dans lequel la probabilité $\text{P}\,(X_t|X_{t-1}, Y_t)$ ne dépend pas de X_{t-1}. Rappelons que dans les modèles à base de filtrage bayésien classique, le modèle des observations $\text{P}\,(X_t|Y_t)$ n'est autre que la vraisemblance des lieux. On peut utiliser cette vraisemblance pour classer les images indépendamment les unes des autres, sans filtrage bayésien. Comme l'ont montré les travaux de Wu, Christensen et Rehg [WCR09] et Torralba et al. [Tor+03] (voir page 141), l'utilisation du filtrage améliore les performances.

La combinaison d'un système de mots visuels définis sur des caractéristiques globales et de filtrage bayésien a été étudiée par [WCR09].

De même, le système BFMC peut fonctionner sans filtrage si l'on ne prend pas en compte l'évolution de l'état. Dans ce cas, un lieu est classé simplement à partir de la vraisemblance $\text{P}\,(X_t|Y_t, X_{1:t-1})$. On retrouve l'algorithme développé dans [Gui09, chap. 8] dans lequel un lieu est reconnu grâce aux transitions entre les prototypes observés. Comme le système tBoW, ce système ne permet pas de prendre en compte les changements de lieu. On doit donc là aussi employer un autre mécanisme pour détecter les transitions ou notifier les transitions au système.

2.3. Apprentissage des paramètres

Dans le cas où l'état et les observations sont des variables discrètes, le modèle est complètement décrit par :

1. la matrice de transition entre les états cachés $\mathbf{A}$ où $\mathbf{A}\,[i,j]$ est la probabilité de passer du lieu c^i au lieu c^j : $\mathbf{A}\,[i,j] = \text{P}\,(Y_t = c^j|Y_{t-1} = c^i)$.
2. la probabilité du premier état $\text{P}\,(Y_0)$; cette distribution est stockée dans un vecteur noté $\boldsymbol{\pi}$: $\boldsymbol{\pi}\,[i] = \text{P}\,(Y_0 = c^i)$.
3. pour chaque lieu c^i, une matrice de transition entre observations $\mathbf{C^i}$ où $\mathbf{C^i}\,[k,l]$ est la probabilité d'observer k suivi de l dans le lieu c^i : $\mathbf{C^i}\,[k,l] = \text{P}\,(X_t = l|X_{t-1} = k, Y_t = c^i)$. Cette matrice constitue donc le modèle de l'apparence du lieu c^i.

Du point de vue de l'apprentissage, la principale difficulté consiste à estimer les matrices $\mathbf{C^i}$. Nous utilisons ici aussi la formule de Laplace-Lidstone (voir Section IV.3.4) :

$$\widehat{\mathbf{C^i}}\,[k,l] = \frac{N^i(k,l) + M}{\sum_{l'} N^i(k,l') + M \times K} \tag{VI.8}$$

où :

- $N^i(k,l)$ est le nombre de transitions du mot k au mot l dans le lieu c^i.
- $\sum_{l'} N^i(k,l')$ est le nombre de fois où l'on a observé k dans le lieu c^i ; ce terme assure la normalisation.

– M est le nombre d'observations virtuelles. Comme dans le chapitre IV, nous utilisons $M = 1$.

En ce qui concerne l'apprentissage de la matrice de transition entre lieux, $\mathbf{A}$, plusieurs approches sont possibles (voir Section III.4.4.1). Comme [WCR09], nous utiliserons une matrice de transition indépendante de la topologie de l'environnement d'apprentissage afin de faciliter la reconnaissance dans de nouveaux environnements :

– On fixe la probabilité p_e de rester dans le même lieu : les éléments diagonaux de $\mathbf{A}$ sont donnés par : $\mathbf{A}[i,i] = p_e$.
– Les élément non-diagonaux sont égaux : leur valeur est donnée par : $\mathbf{A}[i,j] = \frac{1-p_e}{p-1}$.

Pour la probabilité de l'état initial, $\boldsymbol{\pi}$, nous utilisons une distribution uniforme : $\boldsymbol{\pi}[i] = \frac{1}{p}$ (rappelons que p est le nombre de lieux).

On peut faire une analogie entre le modèle BFMC et le problème de l'identification de langue dans lequel on doit reconnaître à quelle langue (parmi un ensemble fini et connu à l'avance) appartient une phrase. La matrice $\mathbf{C^i}$ est similaire aux modèles n-grams utilisés en traitement du langage (avec $n = 2$). En traitement du langage, une langue est caractérisée par la suite de mots qu'elle génère. Par analogie, un lieu est défini par la suite de mots visuels qu'il engendre. La matrice $\mathbf{C^i}$ donne donc la grammaire du lieu. La matrice de transition donne la probabilité de passer d'une langue à l'autre. Cependant, la plupart des études en identification de langue cherchent à classifier des phrases clairement délimitées alors que nous devons fournir une réponse à chaque mot (*i.e.* chaque image).

Afin d'illustrer les performances de l'algorithme BFMC, nous le comparerons, d'une part, à l'algorithme sans filtrage et, d'autre part, au filtrage bayésien classique. Un des avantages du modèle BFMC et plus largement des approches basées sur le filtrage bayésien par rapport au système tBoW, est que la détection des changements de lieux est intégrée au modèle. Nous évaluerons donc les performances de ces algorithmes pour la détection de changements de lieux.

3. Résultats en reconnaissance d'instances

Dans cette partie, nous cherchons à évaluer notre système sur la tâche de reconnaissance d'instances ainsi que l'influence des paramètres les plus importants du système sur cette tâche. Nous nous appuyons sur les résultats du chapitre IV en ne considérant que les signatures CENTRIST et GIST (calculées sur les canaux de couleur et sur la luminance). Les paramètres restants à prendre en compte sont donc :

– La taille de la carte de Kohonen S.
– La valeur de la probabilité de rester dans le même pièce p_e qui définit la matrice de transition.

Comme dans la section IV.4, nous utilisons le taux de reconnaissance moyen correct et le Taux de Vrais Positifs (TVP) moyen sur le protocole de reconnaissance d'instances proposé par [Ull+08] décrit dans la Section A.1.4. Rappelons que ce protocole permet de comparer la réponse du système sous différentes conditions de luminosité.

3.1. Influence de la matrice de transition

Afin d'étudier l'influence de p_e, nous avons répété le protocole pour p_e variant de 0,90 à 0,99 par pas de 0,01. Ici, la taille de la carte de Kohonen est fixée à $S = 20$.

Les résultats sont donnés dans la Figure VI.2. L'augmentation est quasi-linéaire pour tous les descripteurs. Elle est assez faible pour les deux variantes de GIST (passage de 69% à 72% pour les signature sur la luminance et de 73% à 76% pour les signatures sur les canaux de couleur) et un peu plus importante pour CENTRIST (passage de 56% à 64%). La variance augmente fortement pour CENTRIST.

3.2. Influence de la taille de la carte de Kohonen

Afin d'étudier l'influence de la taille de la carte de Kohonen, S, nous avons réalisé l'expérience décrite dans la section section IV.4. Ici, la probabilité de rester dans la même pièce est fixée à $p_e = 0{,}99$.

Les résultats sont donnés dans la Figure VI.3. Les performances (la précision ou le TVP) augmentent avec S. L'effet est à peu près identique pour toutes les signatures.

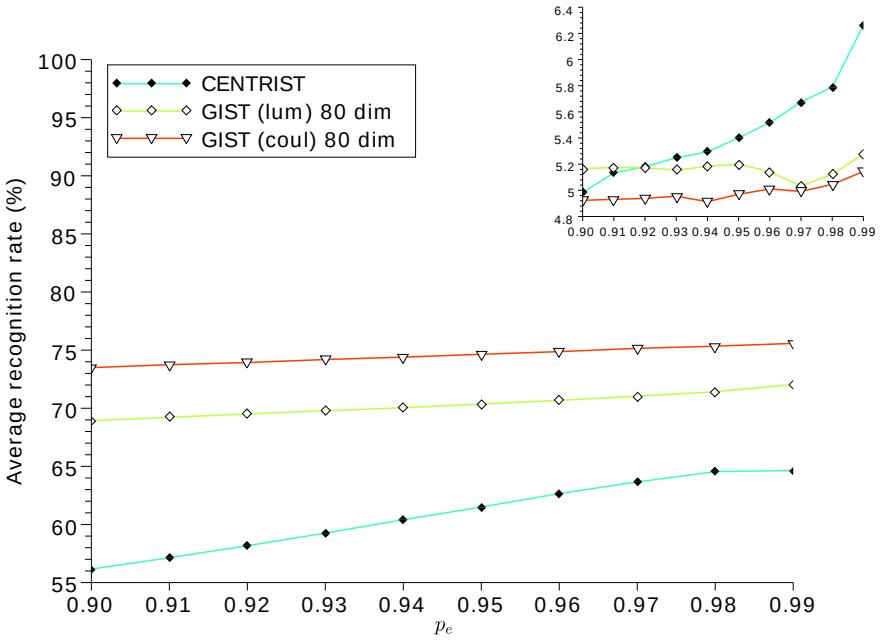
Cet effet est assez surprenant. En effet, chaque matrice $\mathbf{C^i}$ comprend $K^2 = S^4$ entrées. Avec $S = 50$, on doit donc estimer environ 6 millions de termes. On peut essayer d'interpréter cet effet en se souvenant que lorsque l'on utilise des grandes cartes de Kohonen, de nombreuses images successives sont associées à un même prototype (voir Section IV.4.3) et que les prototypes importants se retrouvent dans les séquences de reconnaissance.

3.3. Comparaison au filtrage bayésien classique

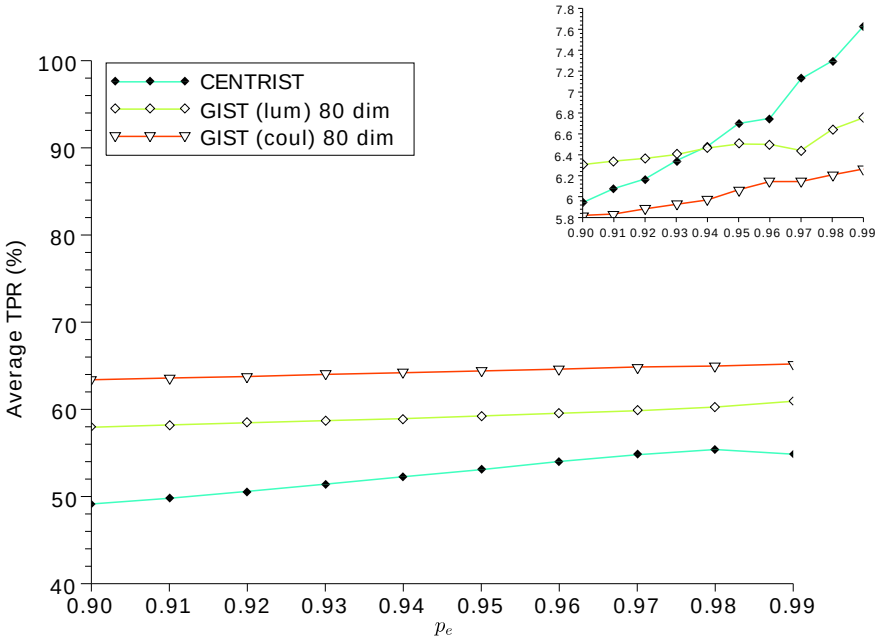
Dans cette section, nous comparons les résultats obtenus sans filtrage, avec le filtrage bayésien classique et avec BFMC. Nous utilisons les paramètres qui ont donné les meilleurs résultats pour BFMC : $p_e = 0{,}99$ et $S = 30$.

Les résultats sont donnés dans les tables VI.1 pour CENTRIST, VI.2 pour le descripteur GIST calculé sur le canal de luminance et VI.3 pour le descripteur GIST calculé sur les canaux de couleurs.

On voit que pour le descripteurs GIST, les résultats sont supérieurs à ceux du filtrage bayésien classique. Pour le descripteur CENTRIST cependant, les résultats de BFMC sont

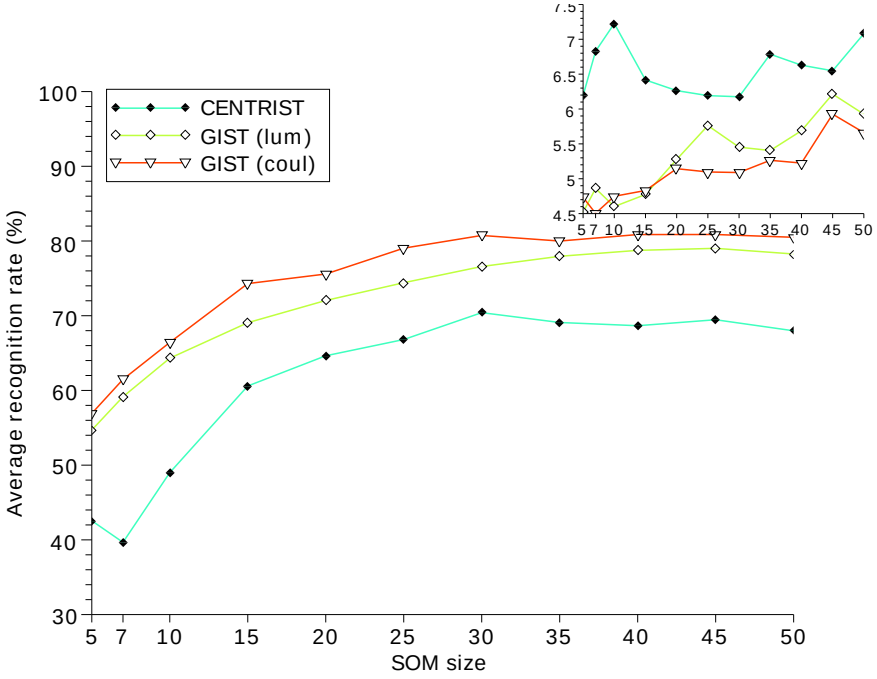


(a) Taux de reconnaissance moyen

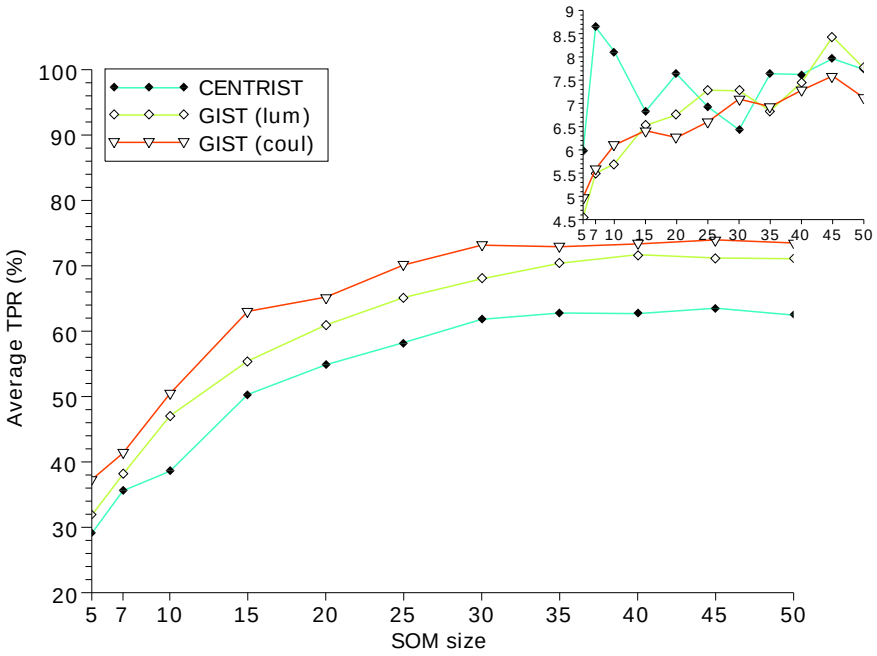


(b) TVP

FIGURE VI.2.: Influence de p_e en reconnaissance d'instances : (a) sur le taux de reconnaissance moyen, (b) sur le TVP moyen.



(a) Taux de reconnaissance moyen



(b) TVP

FIGURE VI.3.: Influence de S en reconnaissance d'instances : (a) sur le taux de reconnaissance moyen, (b) sur le TVP moyen.

	Pas de filtrage	Filtrage bayésien	BFMC
Saarbruecken-Std	$36{,}20\%^{\pm 3{,}21\%}$	$86{,}00\%^{\pm 2{,}98\%}$	$82{,}45\%^{\pm 4{,}12\%}$
Saarbruecken-Ext	$30{,}54\%^{\pm 3{,}45\%}$	$78{,}45\%^{\pm 5{,}41\%}$	$75{,}25\%^{\pm 6{,}05\%}$
Freiburg-Std	$32{,}66\%^{\pm 3{,}49\%}$	$81{,}13\%^{\pm 4{,}65\%}$	$75{,}04\%^{\pm 6{,}52\%}$
Freiburg-Ext	$29{,}95\%^{\pm 2{,}30\%}$	$75{,}10\%^{\pm 3{,}07\%}$	$68{,}62\%^{\pm 3{,}80\%}$
Ljubljana-Std	$23{,}19\%^{\pm 2{,}39\%}$	$66{,}80\%^{\pm 6{,}50\%}$	$56{,}73\%^{\pm 9{,}63\%}$
Ljubljana-Ext	$21{,}15\%^{\pm 1{,}95\%}$	$62{,}93\%^{\pm 6{,}24\%}$	$54{,}82\%^{\pm 7{,}75\%}$
Average	$28{,}95\%^{\pm 2{,}80\%}$	$75{,}07\%^{\pm 4{,}81\%}$	$68{,}82\%^{\pm 6{,}31\%}$

TABLE VI.1.: Résultats moyens pour le protocole de reconnaissance d'instances sans filtrage, avec le filtrage bayésien classique et avec BFMC pour le descripteur CENTRIST.

moins bons que ceux du filtrage bayésien classique. La séquence des prototypes d'entrée varie plus avec CENTRIST, ce qui est confirmé par les très faibles résultats sans filtrage (28 % contre largement plus de 50 % pour GIST). Ceci explique que la prise en compte des dépendances entre images n'améliore pas les résultats par rapport au filtrage bayésien classique.

3.4. Détection des transitions

Nous présentons ici les résultats obtenus, d'une part, par le filtrage bayésien et, d'autre part, par BFMC sur la tâche de détection des transitions. Le paramètre p_e est fixé à 0,99 et S est fixé 20, la même valeur que dans le Section V.4.1 pour faciliter les comparaisons.

Les résultats sont donnés dans la Table VI.4. Comparé au filtrage bayésien, BFMC détecte beaucoup moins de transitions, ce qui se traduit en général par un TVP plus faible mais aussi un Taux de Faux Positifs (TFP) plus faible. Les résultats en reconnaissance d'instances sont meilleurs avec BFMC, sauf pour CENTRIST.

Comparé aux résultats de la Section V.4.1, BFMC donne de meilleurs résultats pour GIST (75,58 % contre 70,60 %). Pour CENTRIST en revanche, les résultats sont inférieurs (64,64 % contre 71,50 %). Le filtrage bayésien classique donne des résultats très proches de tBoW pour CENTRIST mais inférieurs pour CENTRIST. Notons toutefois que, contrairement à l'algorithme tBoW, nous n'avons pas à introduire de paramètres supplémentaires pour détecter les transitions.

	Pas de filtrage	**Filtrage bayésien**	**BFMC**
Saarbruecken-Std	$55{,}49\%^{\pm 4{,}95\%}$	$78{,}33\%^{\pm 4{,}89\%}$	$81{,}29\%^{\pm 4{,}56\%}$
Saarbruecken-Ext	$43{,}49\%^{\pm 5{,}09\%}$	$71{,}14\%^{\pm 7{,}97\%}$	$72{,}76\%^{\pm 7{,}74\%}$
Freiburg-Std	$52{,}52\%^{\pm 3{,}97\%}$	$71{,}26\%^{\pm 7{,}02\%}$	$76{,}32\%^{\pm 5{,}90\%}$
Freiburg-Ext	$38{,}99\%^{\pm 2{,}98\%}$	$63{,}28\%^{\pm 4{,}15\%}$	$68{,}45\%^{\pm 4{,}59\%}$
Ljubljana-Std	$63{,}41\%^{\pm 3{,}68\%}$	$63{,}32\%^{\pm 8{,}33\%}$	$82{,}97\%^{\pm 5{,}03\%}$
Ljubljana-Ext	$57{,}84\%^{\pm 2{,}27\%}$	$62{,}67\%^{\pm 6{,}93\%}$	$77{,}48\%^{\pm 4{,}08\%}$
Average	$51{,}96\%^{\pm 3{,}82\%}$	$68{,}33\%^{\pm 6{,}55\%}$	$76{,}54\%^{\pm 5{,}32\%}$

TABLE VI.2.: Résultats moyens pour le protocole de reconnaissance d'instances sans filtrage, avec le filtrage bayésien classique et avec BFMC pour le descripteur GIST calculé sur le canal de luminance.

	Pas de filtrage	**Filtrage bayésien**	**BFMC**
Saarbruecken-Std	$62{,}61\%^{\pm 4{,}32\%}$	$82{,}43\%^{\pm 4{,}08\%}$	$85{,}75\%^{\pm 3{,}78\%}$
Saarbruecken-Ext	$50{,}80\%^{\pm 5{,}56\%}$	$76{,}61\%^{\pm 7{,}80\%}$	$78{,}86\%^{\pm 7{,}18\%}$
Freiburg-Std	$58{,}10\%^{\pm 5{,}35\%}$	$81{,}02\%^{\pm 5{,}26\%}$	$84{,}69\%^{\pm 4{,}52\%}$
Freiburg-Ext	$46{,}76\%^{\pm 3{,}06\%}$	$70{,}48\%^{\pm 3{,}37\%}$	$75{,}90\%^{\pm 3{,}90\%}$
Ljubljana-Std	$71{,}49\%^{\pm 3{,}72\%}$	$57{,}08\%^{\pm 9{,}75\%}$	$80{,}42\%^{\pm 6{,}02\%}$
Ljubljana-Ext	$63{,}95\%^{\pm 3{,}29\%}$	$57{,}07\%^{\pm 9{,}03\%}$	$75{,}69\%^{\pm 5{,}03\%}$
Average	$58{,}95\%^{\pm 4{,}22\%}$	$70{,}78\%^{\pm 6{,}55\%}$	$80{,}22\%^{\pm 5{,}07\%}$

TABLE VI.3.: Résultats moyens pour le protocole de reconnaissance d'instances sans filtrage, avec le filtrage bayésien classique et avec BFMC pour le descripteur GIST calculé sur les canaux de couleur.

Intégration temporelle	Descripteur	# détections	TVP	TFP	% correct
Filtrage bayésien	CENTRIST	13 121	63,65 %	92,54 %	70,99 %
	GIST (couleur)	10 099	67,04 %	89,79 %	67,97 %
	GIST (luminance)	13 120	75,36 %	91,17 %	63,78 %
BFMC	CENTRIST	3 714	33,16 %	86,27 %	64,64 %
	GIST (couleur)	5 197	59,36 %	82,43 %	75,58 %
	GIST (luminance)	6 102	63,00 %	84,12 %	72,04 %

TABLE VI.4.: Comparaison du filtrage bayésien et de BFMC pour la détection des transitions.

Globalement, le TFP reste encore élevé mais BFMC semble plus avantageux que le filtrage bayésien classique et que la détection des transitions avec tBoW.

3.5. Comparaison avec l'état de l'art et conclusion

En ce qui concerne les descripteurs, les différentes variantes de GIST ont de meilleurs résultats que CENTRIST. Nous avons souligné que cela est probablement du au fait que, avec CENTRIST, des images proches temporellement peuvent se retrouver sur des prototypes différents. C'est pourquoi la prise des corrélations entre images n'améliore pas les résultats avec ce descripteur. La variante de GIST basée sur les canaux de couleur donne de meilleurs résultats.

Globalement, les systèmes présentés ici ont des performances inférieures à celles de [Ull+08], bien que dans le cas du systèmes utilisant GIST calculé sur les canaux de couleur, les différences ne soient pas significatives (nous sommes donc au niveau de l'état de l'art). Les différences sont en revanche significatives avec le système tBoW lorsque l'on notifie les vrais changements de lieux.

En ce qui concerne CENTRIST, les performances sont du même ordre que celles du système tBoW lorsque l'on utilise la détection des seuils (bien que la différence soit statistiquement significative). Les variantes de GIST ont des performances supérieures au système tBoW lorsque l'on utilise la détection des seuils.

Globalement, BFMC atteint des performances comparables à l'état de l'art pour la tâche de reconnaissance d'instances.

4. Résultats en catégorisation

Dans cette partie, nous cherchons à évaluer notre système sur la tâche de catégorisation de lieux sémantiques ainsi que l'influence des mêmes paramètres sur cette tâche. Comme dans la section IV.5, nous utilisons le protocole de catégorisation décrit dans la Section A.1.5.

4.1. Influence de la matrice de transition

Nous étudions ici l'influence de la valeur p_e sur le TVP moyen. Ici, la taille de la carte de Kohonen est fixée à $S = 20$. Nous avons répété le protocole pour p_e variant de 0,90 à 0,99 par pas de 0,01.

Les résultats sont donnés dans la Figure VI.4. Les résultats augmentent légèrement quand p_e augmente. Ici aussi, l'effet est faible pour la signature GIST calculée sur le canal de luminance. Il est un peu plus important pour la signature GIST calculée sur les canaux de couleurs et sur CENTRIST.

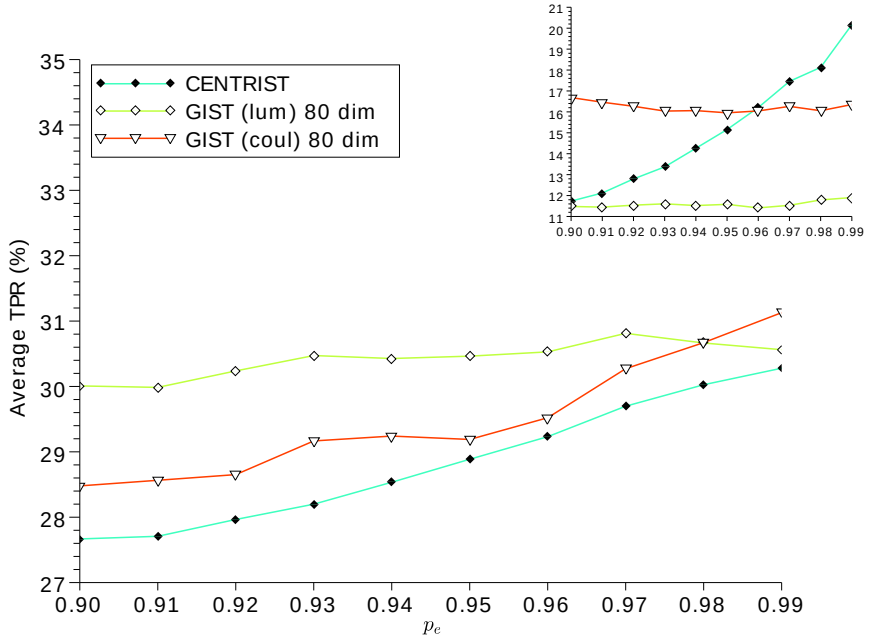


FIGURE VI.4.: Influence de p_e sur le TVP moyen en catégorisation.

4.2. Influence de la taille de la carte de Kohonen

Pour étudier l'influence de la taille de la carte de Kohonen, nous avons répété le protocole pour les valeurs de S données dans la Table IV.2 (voir page 167). Ici, la probabilité de rester dans la même pièce est fixée à $p_e = 0{,}99$.

Les résultats sont donnés dans la Figure VI.5 :

- Pour CENTRIST, les performances augmentent jusqu'à $S = 30$ puis diminuent.
- Pour la signature GIST calculée sur le canal de luminance, les performances augmentent continument.
- Pour la signature GIST calculée sur les canaux de couleur, les performances varient fortement pour les tailles supérieures à 30.

4.3. Comparaison au filtrage bayésien classique

Nous comparons ici les résultats de l'algorithme BFMC avec ceux obtenus sans filtrage et avec le filtrage bayésien classique. Nous utilisons les signatures qui ont donné les meilleurs résultats :

- pour le descripteur GIST calculé sur le canal de luminance : $p_e = 0{,}99$ et $S = 35$.

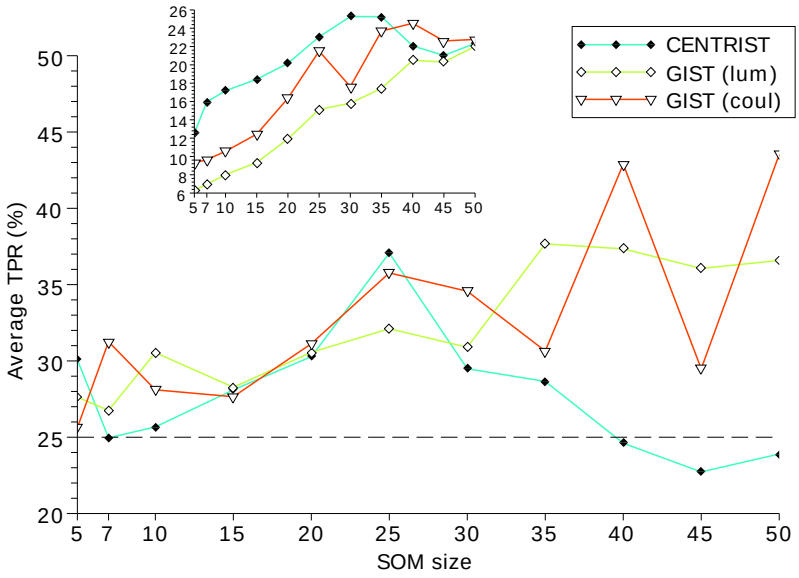


FIGURE VI.5.: Influence de S sur le TVP moyen en catégorisation.

– pour le descripteur GIST calculé sur les canaux de couleur : $p_e = 0{,}99$ et $S = 40$.
– pour le descripteur CENTRIST : $p_e = 0{,}99$ et $S = 25$.

Le TVP moyen sur tout l'expérience est présenté dans la Table VI.5. On voit que pour toutes les signatures, BFMC augmente significativement les performances par rapport à l'absence de filtrage et par rapport au filtrage bayésien classique.

	Pas de filtrage	**Filtrage bayésien**	**BFMC**
GIST (luminance)	26,66 %	36,46 %	37,67 %
GIST (couleur)	24,96 %	27,55 %	42,90 %
CENTRIST	25,01 %	33,43 %	37,07 %

TABLE VI.5.: Résultats moyens pour le protocole de catégorisation sans filtrage, avec le filtrage bayésien classique et avec BFMC.

Intégration temporelle	Descripteur	# détections	TVP	TFP	% correct
Filtrage bayésien	CENTRIST	8 680	45,93 %	97,86 %	32,66 %
	GIST (couleur)	2 244	10,12 %	98,17 %	26,02 %
	GIST (luminance)	5 245	36,54 %	97,18 %	36,46 %
BFMC	CENTRIST	2 245	16,54 %	97,02 %	28,66 %
	GIST (couleur)	1 178	1,98 %	99,32 %	30,65 %
	GIST (luminance)	1 751	7,65 %	98,23 %	37,67 %

TABLE VI.6.: Comparaison du filtrage bayésien et de BFMC pour la détection des transitions en catégorisation.

4.4. Détection des transitions

Nous présentons ici les résultats obtenus, d'une part, par le filtrage bayésien et, d'autre part, par BFMC sur la tâche de détection des transitions. Le paramètre p_e est fixé à 0,99 et S est fixé 35, la même valeur que dans la Section V.4.2 pour faciliter les comparaisons. Notons que la valeur de S est loin d'être optimale pour BFMC (voir Figure VI.5) alors qu'elles était plutôt bonne pour tBoW.

Les résultats sont donnés dans la Table VI.6. Comparé au filtrage bayésien, ici aussi, BFMC détecte beaucoup moins de transitions, ce qui se traduit en général par un TVP plus faible mais aussi un TFP plus faible. Les résultats en reconnaissance d'instances sont meilleurs avec BFMC, sauf pour CENTRIST.

Comparé aux résultats de la Section V.4.2, les résultats en catégorisation avec BFMC sont du même ordre qu'avec tBoW pour GIST mais inférieurs pour CENTRIST (il faudrait tester des valeurs de S différentes). Notons toutefois que le TVP est en général plus faible avec BFMC (mais il y a moins de détections). Pour le filtrage bayésien, les résultats sont comparables à ceux de tBoW.

Globalement, les performances en catégorisation étant toujours relativement assez basses, la détection des transitions est toujours difficile. Notons que nos critères d'évaluations favorisent mécaniquement un système qui détecte beaucoup de transitions. Augmenter le seuil Δt conduirait sans doute à des résultats plus avantageux pour BFMC.

4.5. Comparaison à l'état de l'art et conclusion

4.5.1. Comparaison sur la base COLD

Les performances en catégorisation sont illustrés dans la Figure VI.6 avec les paramètres utilisés dans la section précédente :

– avec le descripteur GIST calculé sur le canal de luminance, le TVP moyen est de 37,67%.
– avec le descripteur GIST calculé sur les canaux de couleur, le TVP moyen est de 42,90%.
– avec le descripteur CENTRIST, le TVP moyen est de 37,08%.

Globalement, ces résultats sont supérieurs à ceux obtenus avec l'algorithme tBoW (voir section IV.5).

On note encore une fois que certaines catégories sont mieux reconnues que d'autres. Cependant, comparé au système tBoW (voir Figure IV.12, page 183), le TVP semble moins varier entre les pièces. En particulier, il y a moins de cas où le TVP est nul ou très faible. Notons que, comme dans le chapitre IV, les performances d'un descripteur pour une catégorie donnée varient fortement entre les laboratoires et les conditions de luminosité. Ainsi, le descripteur GIST calculé sur les canaux de couleurs donne de bonnes performances pour les toilettes la nuit mais des performances nettement moins bonnes pour les jours nuageux (voir VI.6b).

Encore une fois, GIST donne de meilleures performances que CENTRIST, surtout lorsque l'on utilise les canaux de couleurs. Cette importance des couleurs dans la tâche de catégorisation est surprenante au vu de l'état de l'art et du chapitre IV. Plus d'études seraient nécessaires pour tester si cet effet est robuste, en particulier vis-à-vis de la taille de la carte auto-organisatrice.

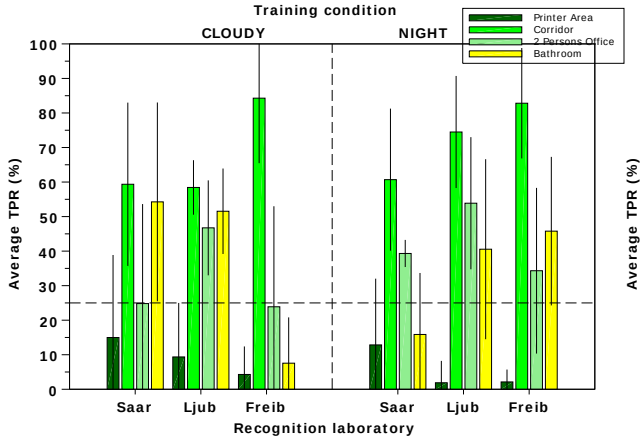
4.5.2. Comparaison sur la base VPC

Les tables VI.7 montrent les performances du système sur le protocole de catégorisation décrit dans la Section A.2.3. Les résultats sont très inférieurs à ceux obtenus par Wu [Wu09] et Ranganathan [Ran10], ce qui est surprenant au vu des bonnes performances sur la base COLD. Un réglage plus soigneux des paramètres permettrait sans doute de meilleures performances.

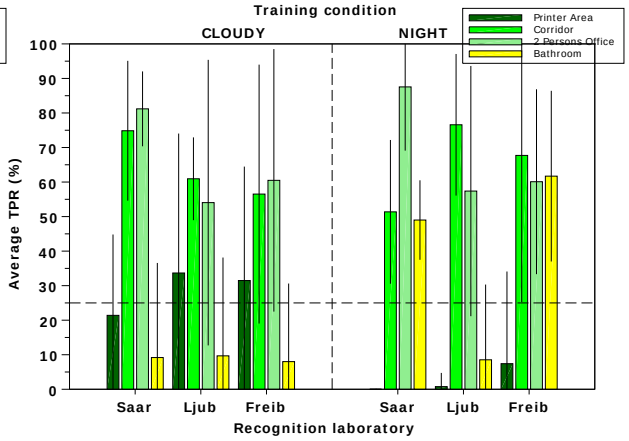
Le système BFMC semble donc bien apporter une amélioration sur la tâche de catégorisation, mais l'effet semble assez sensible aux paramètres.

5. Conclusion

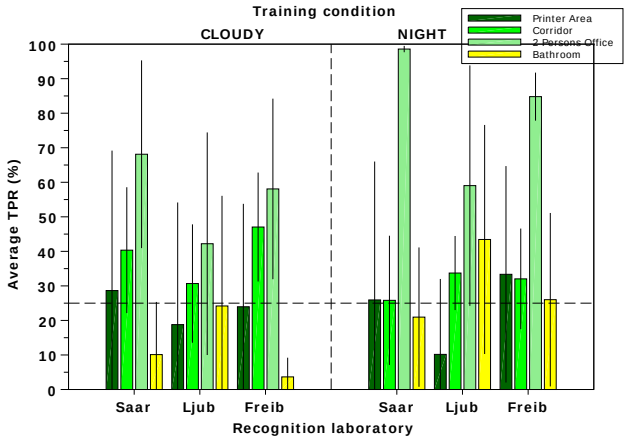
Nous avons présenté dans ce chapitre un système de reconnaissance des lieux basé sur les même mots et les mêmes signatures que le système tBoW mais qui essaye de dépasser l'hypothèse d'indépendance des observations, sans toutefois modéliser toute la séquence d'apparition des mots.



(a) GIST sur canal de luminance



(b) GIST sur canaux de couleur



(c) CENTRIST

FIGURE VI.6.: Exemples de résultats en catégorisation avec BFMC : (a) avec le descripteur GIST sur les canaux de couleur, (b) avec le descripteur GIST sur les canaux de couleur, (c) avec le descripteur CENTRIST. Les barres d'erreur représentent l'écart-type. La condition d'illumination utilisée pour l'entraînement et la reconnaissance est montrée sur l'axe horizontal haut. Le laboratoire utilisé pour la reconnaissance est indiqué en dessous. La ligne horizontale en pointillés indique la ligne de chance.

	Categories					
Home	**Bed**	**Bath**	**Kitchen**	**Living**	**Dining**	**Average**
1	62,88 %	44,93 %	30,71 %	0,00 %	8,85 %	29,47 %
2	48,67 %	19,06 %	45,34 %	16,10 %	0,00 %	25,84 %
3	53,22 %	70,59 %	20,70 %	1,58 %	0,00 %	29,22 %
4	53,58 %	12,74 %	43,50 %	30,66 %	0,74 %	28,24 %
5	44,07 %	48,50 %	10,61 %	0,00 %	3,59 %	21,35 %
6	50,06 %	26,51 %	31,06 %	0,00 %	14,67 %	24,46 %
Average	52,08 %	37,05 %	30,32 %	8,06 %	4,64 %	26,43 %

TABLE VI.7.: Performances obtenues sur la base VPC pour le descripteur CENTRIST ($S = 25$, $p_e = 0{,}99$).

	Categories					
Home	**Bed**	**Bath**	**Kitchen**	**Living**	**Dining**	**Average**
1	57,10 %	27,71 %	13,07 %	52,44 %	8,20 %	31,70 %
2	51,18 %	31,10 %	7,09 %	1,13 %	9,89 %	20,08 %
3	59,42 %	27,00 %	24,39 %	13,97 %	1,97 %	25,35 %
4	43,68 %	47,88 %	20,63 %	10,11 %	22,92 %	29,04 %
5	44,29 %	58,66 %	11,20 %	8,52 %	1,64 %	24,86 %
6	46,07 %	43,76 %	14,68 %	2,37 %	6,22 %	22,62 %
Average	50,29 %	39,35 %	15,18 %	14,76 %	8,48 %	25,61 %

TABLE VI.8.: Performances obtenues sur la base VPC pour le descripteur GIST calculé sur le canal de luminance ($S = 35$, $p_e = 0{,}99$).

	Categories					
Home	**Bed**	**Bath**	**Kitchen**	**Living**	**Dining**	**Average**
1	51,52 %	33,87 %	32,78 %	23,17 %	23,61 %	32,99 %
2	63,74 %	34,04 %	35,43 %	17,23 %	13,07 %	32,70 %
3	31,47 %	26,40 %	14,04 %	2,21 %	5,01 %	15,82 %
4	30,43 %	29,83 %	53,81 %	14,77 %	12,94 %	28,36 %
5	37,72 %	43,20 %	3,46 %	22,81 %	8,82 %	23,20 %
6	43,33 %	29,73 %	6,60 %	33,56 %	0,00 %	22,64 %
Average	43,03 %	32,85 %	24,35 %	18,96 %	10,57 %	25,95 %

TABLE VI.9.: Performances obtenues sur la base VPC pour le descripteur GIST calculé sur les canaux de couleur ($S = 40$, $p_e = 0{,}99$).

Globalement, les résultats en reconnaissance d'instances sont au niveau de l'état de l'art. En catégorisation, nous observons un gain intéressant sur certaines bases mais le Taux de Vrais Positifs reste quand même assez bas par rapport à la reconnaissance d'instances.

Plusieurs pistes d'améliorations sont possibles. Tout d'abord, nous avons souligné que le modèle d'apparence du lieu est similaire aux modèles n-grams en traitement de la langue. Ces modèles ont été intensément étudiés et des méthodes d'estimation plus efficaces ont été proposées (voir [MS99]). De plus, nous avons souligné que l'ordre local est lié au fait que la géométrie des pièces contraint les actions du robot. Dans cette optique, on pourrait prendre en compte les actions du robot ce qui est possible dans le formalisme bayésien.

Bibliographie

[BBR10] C. Barber, J. Bockhorst et P. Roebber. « Auto-Regressive HMM Inference with Incomplete Data for Short-Horizon Wind Forecasting ». Dans : *Advances in Neural Information Processing Systems 23*. Sous la dir. de J. Lafferty et al. 2010, p. 136–144 (cf. p. 217).

[Ber99] A. Berchtold. « The double chain Markov model ». Dans : *Communications in Statistics - Theory and Methods* 28.11 (1999), p. 2569–2589 (cf. p. 217).

[ER05] Y. Ephraim et W. Roberts. « Revisiting autoregressive hidden Markov modeling of speech signals ». Dans : *IEEE Signal Processing Letters* 12.2 (2005), p. 166–169 (cf. p. 217).

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 215, 218).

[MS99] C. Manning et H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999 (cf. p. 232).

[Ran10] A. Ranganathan. « PLISS : Detecting and Labeling Places using Online Change-point Detection ». Dans : *Proceedings of the 2010 Robotics : Science and Systems Conference (RSS 2010)*. 2010 (cf. p. 229).

[Tan04] X. Tang. « Autoregressive hidden markov model with application in an El Niño study ». Mém.de maîtr. University of Saskatchewan, 2004 (cf. p. 217).

[Tor+03] A. Torralba et al. *Context-based vision system for place and object recognition*. Rap. tech. Cambridge, MA, 2003 (cf. p. 218).

[Ull+08] M. M. Ullah et al. « Towards Robust Place Recognition for Robot Localization ». Dans : *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008)*. Pasadena, USA, mai 2008 (cf. p. 220, 225).

[WCR09] J. Wu, H. Christensen et J. Rehg. « Visual place categorization : Problem, dataset, and algorithm ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE. St. Louis, USA, 2009, p. 4763–4770 (cf. p. 218, 219).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 229).

VII. Conclusion et perspectives

1. Contributions

Nous avons présenté et évalué deux modèles génératifs et incrémentaux de reconnaissance de lieux sémantiques, appelés *temporal Bag-of-Words (tBoW)* et *Bayesian Filtering with Markov Chains (BFMC)*. Ces deux méthodes sont basées sur des mots visuels, obtenus par un processus de classification non-supervisée à partir de différentes caractérisations globales de l'image. L'algorithme de classification non-supervisée utilisé est l'algorithme de Kohonen.

Considérés indépendamment, ces mots visuels ne permettent pas une reconnaissance correcte des lieux. L'idée générale de notre travail est donc de prendre en compte plusieurs images pour améliorer la classification. Classiquement, on utilise dans ce cas le formalisme du filtrage bayésien dans lequel le rapport entre les observations et les lieux est modélisé par un Modèle de Markov Caché (HMM).

Dans le système tBoW, issu des travaux de thèse de Hervé Guillaume, un lieu est reconnu à partir d'une fenêtre contenant les L dernières observations. Le filtrage bayésien est simplifié, d'une part, en ne tenant pas compte des changements de lieux et, d'autre part, en supposant que les observations sont indépendantes, ce qui conduit à une règle d'intégration temporelle très simple. L'apprentissage est basé sur un modèle génératif et incrémental. Nous avons montré que, si l'on suppose possible de re-initialiser la fenêtre lors des changements de lieu, les performances sont largement supérieures à l'état de l'art. Une étude paramétrique nous a permis de montrer que, pour la tâche de reconnaissance d'instances, plus on utilise d'observations en reconnaissance plus les performances augmentent. En catégorisation, il est préférable d'utiliser des fenêtres plus petites contenant quelques dizaines d'images.

Logiquement, l'étape suivante était de chercher un système capable de détecter lui-même les changements de lieux, ce qui est une nouveauté par rapport à [Gui09]. Nous avons montré que l'entropie de la classification obtenue présentait des variations statistiquement importantes au voisinage des changements de lieux. L'utilisation de ces changements pour détecter les transitions conduit cependant à un nombre important de faux positifs, ce qui réduit la taille de la fenêtre effectivement utilisée par le système. Les performances sont plutôt bonnes en reconnaissance d'instances mais toutefois inférieures au cas idéal. En catégorisation, nous obtenons parfois de meilleures performances en re-initialisant souvent la fenêtre.

Dans le système BFMC, le filtrage bayésien est enrichi en utilisant la corrélation entre observations. Nous avons montré que, pour certaines signatures, cette règle est plus efficace

que le filtrage bayésien classique, sans engendrer des coûts d'apprentissage prohibitifs. Nous avons aussi montré que les performances sont supérieures à celles du modèle tBoW (sauf si l'on suppose que la détection des seuils est parfaite). De plus, le système BFMC est au niveau de l'état de l'art sur les tâches de reconnaissance d'instances et apporte une amélioration sur la tâche de catégorisation. Ce système constitue donc une avancée par rapport aux travaux de [Gui09].

Ces résultats démontrent que l'on peut catégoriser et reconnaître des lieux de manière robuste à partir d'une information fruste, constituée de mots visuels, à condition d'utiliser une méthode d'intégration temporelle, c'est-à-dire de prendre en compte plusieurs images.

Toutes les méthodes que nous avons présentées peuvent fonctionner en ligne et de manière incrémentale, la seule restriction étant l'entraînement de la carte de Kohonen. D'un point de vue computationnel, les méthodes sont extrêmement légères, la phase la plus gourmande étant le calcul des signatures (celles-ci étant relativement simples par rapport à des descriptions plus poussées de l'image).

Nos travaux ont également permis de comparer en profondeur plusieurs signatures globales, notamment leur robustesse aux changements dynamiques et aux changements de luminosité. À notre connaissance, c'est la première fois que GIST et CENTRIST sont comparés dans un système basé sur les mots visuels. Le descripteur GIST s'est en général montré plus performant que CENTRIST. Wu, Christensen et Rehg [WCR09] avait abouti à la conclusion inverse sur une tâche de reconnaissance de scènes mais n'utilisaient ni mots visuels, ni intégration temporelle. On peut donc supposer que GIST bénéficie plus de ces techniques. Cependant, la comparaison n'est pas tout à fait équitable car d'une part, dans nos travaux, la signature CENTRIST ne prend pas en compte la géométrie grossière de l'image et, d'autre part, le descripteur CENTRIST étant de dimension bien plus grande, il nécessite sans doute plus de clusters. Quoi qu'il en soit, le descripteur CENTRIST conserve un intérêt car il a montré la meilleure robustesse aux changements de luminosité. Notons qu'aucun descripteur ne donne de bonnes performances pour toutes les catégories de lieux et tous les environnements.

Nous avons montré que le nombre de clusters a une grande influence sur les performances : l'utilisation d'un grand nombre de clusters est bénéfique pour la reconnaissance d'instances ; en ce qui concerne la catégorisation, les résultats sont moins clairs et nécessiteraient plus d'investigations.

De manière générale, la tache de reconnaissance d'instances est bien maîtrisée. En revanche, la tâche de catégorisation est elle toujours ouverte. En particulier dans nos expériences, les performances obtenues sur cette tâche étaient très sensibles aux paramètres.

2. Perspectives

On peut, dans un premier temps, envisager de nombreuses améliorations de notre approche, que nous détaillons dans la Section VII.2.1. De plus, on peut, avec le recul, essayer d'envisager des travaux à plus long terme, que nous esquissons dans la Section VII.2.2.

2.1. Améliorations du système

Dans l'immédiat, nous pouvons envisager trois pistes d'amélioration de nos travaux, que nous avons commencé à explorer. Ces améliorations touchent les principaux composants du système, à savoir les images d'entrées, la constitution du vocabulaire visuel et les modèles des lieux.

2.1.1. Sélection des images

Nous avons souligné que, du fait du mouvement du robot, certaines images sont floues. Dans notre approche actuelle, le système doit donner une réponse pour chaque image. Si la qualité de l'image d'entrée est faible, on risque de dégrader les classifications.

Il existe des méthodes simples et applicables en temps réel (voir [RMM10]), pour détecter les images floues. On pourrait donc utiliser ces méthodes comme un filtre : seules les images de qualité suffisante seraient classifiées.

Au-delà, on peut essayer de ne caractériser qu'une sous-partie de l'image. Le module attentionnel développé par Hervé Guillaume durant sa thèse permettant d'extraire des zones variant peu entre les différentes images, peut être considéré comme un premier pas dans cette direction.

Il est tentant d'essayer de supprimer les images contenant le moins d'information *a priori*, par exemple celles obtenues lorsque le robot se trouve face à un mur. Ici aussi, l'idée est qu'il vaut mieux ne pas classer certaines images. Un critère simple consisterait à ne retenir que les images dont l'entropie dépasse une certaine valeur. Cependant, comme nous l'avons dit, il est *a priori* difficile de savoir si une image sera discriminante ou non. il est donc possible au contraire que cette voie mène à une diminution des performances. En revanche, une fois les modèles de lieux appris, on peut rejeter les images peu discriminantes en se basant sur un test statistique.

Il reste à déterminer si ces techniques sont utiles.

2.1.2. Compréhension du rôle de la carte de Kohonen et amélioration des mots visuels

L'algorithme d'obtention des mots visuels joue évidemment un rôle clé dans les performances puisqu'il permet de réduire la variabilité perceptuelle sans introduire de confusion. Nous

avons vu que le rôle de la carte de Kohonen dans les performances pouvait être difficile à interpréter. Ainsi, il semble nécessaire de faire plus d'expériences dans ce domaine.

Tout d'abord la plupart des algorithmes de quantification vectorielle sont sensibles à l'initialisation. Ainsi, il serait intéressant de tester nos systèmes avec plusieurs cartes afin de contrôler cette variabilité. Concernant l'algorithme des $K - moyennes$, il existe des variantes plus robustes à l'initialisation ([AV07]), dont nous pourrions nous inspirer pour rendre l'entraînement des cartes de Kohonen plus robuste.

Les cartes de Kohonen possèdent un avantage majeur : des cellules proches sur la carte sont en général proches dans l'espace d'entrée. Il est dès lors facile de construire des partitions au-dessus de la partition réalisée par les cellules de la carte. L'algorithme de [VS02] est représentatif de ce travail : il permet de regrouper des cellules proches sur la carte et dans l'espace d'entrée en un cluster de plus haut-niveau. On peut dès lors utiliser ces clusters pour définir des mots. L'idée est qu'un cluster de plus haut niveau va regrouper davantage d'images et donc diminuer la variabilité perceptuelle. Néanmoins, les expériences préliminaires que nous avons pu mener n'ont pas montré d'améliorations.

Une autre piste serait d'employer des distances plus spécifiques aux descripteurs que nous utilisons (par exemple les distances spécifiques aux histogrammes pour CENTRIST).

Enfin, il serait aussi intéressant d'utiliser des algorithmes de quantification vectorielle plus récents tels que *Affinity Propagation* (voir [FD07]).

2.1.3. Améliorations de l'apprentissage des modèles de lieux

Du point de vue de l'apprentissage, nous pourrions essayer d'employer des méthodes d'estimation des densités de probabilités plus robustes, en particulier pour le système BFMC. On pourrait aussi étudier l'effet des méthodes de sélection des mots pertinents et des schémas de pondération (même si [Tir10] a montré que ce n'est pas pertinent pour d'autres tâches).

Toujours du point de vue de l'apprentissage, nous avons vu qu'aucun descripteur ne donne de bonnes performances pour toutes les catégories de lieux. Il serait intéressant de combiner plusieurs classificateurs afin d'utiliser les complémentarités de ces descripteurs.

2.2. Travaux futurs

Nous proposons deux grandes pistes de recherche pour la reconnaissance de lieux sémantiques. De manière générale, les lieux sémantiques reposent sur une discrétisation de l'environnement pertinente pour un humain, mais très difficile à atteindre pour un robot. En effet, pour un être humain, la catégorisation fait appel à des connaissances cognitives comme l'agencement de la pièce, une mémoire des situations rencontrées, *etc.*

Pour clore ce gouffre sémantique, les travaux récents en cartographie sémantique ([Pro11]) utilisent des ontologies, c'est-à-dire une représentation formelle des connaissances humaines. Bien qu'elles contiennent des éléments adaptatifs, ces ontologies ne sont, en général, pas

apprises par le robot, ce qui constitue une limite forte à ces travaux d'un point de vue fondamental.

On pourrait essayer de chercher à catégoriser les lieux sémantiques du point de vue du robot. Pour que celui-ci puisse forger les catégories fonctionnelles pertinentes de son point de vue, il est nécessaire qu'il puisse interagir avec l'environnement. Il n'est pas sûr que ces catégories fonctionnelles puissent être mises en correspondance avec les catégories humaines. On touche là un domaine plus général que la cartographie sémantique.

Sans aller aussi loin, nous proposons, pour essayer de clore le gouffre sémantique, d'un côté d'ancrer la reconnaissance de lieux dans le processus d'exploration de l'environnement et de l'autre, d'extraire des informations plus pertinentes à partir des perceptions du robot.

2.2.1. Navigation active et reconnaissance de lieux

Comme nous l'avons dit, la reconnaissance de lieux sémantiques peut servir à guider des stratégies de navigation de haut niveau. En retour, la navigation permet d'obtenir une carte, ce qui peut nous aider par exemple à détecter les transitions. La discrétisation opérée par les cartes topologiques pourrait nous servir à définir les lieux sémantiques à partir d'informations dépendantes des capacités sensori-motrices du robot.

Les techniques de navigation active, dans lesquelles le robot cherche à maximiser l'exploration de l'espace, peuvent aussi permettre d'envisager de coupler ces deux tâches.

L'utilisation d'un modèle de filtrage bayésien comprenant les actions du robot serait un premier pas dans ce sens. Une autre piste serait de modifier la matrice de probabilité de changement de pièce en fonction du temps. En effet, une fois entré dans un lieu, la probabilité de passer à un autre est quasi-nulle pendant une durée assez longue.

2.2.2. Propriétés perceptuelles des lieux

La plupart des méthodes utilisées en reconnaissance de lieux sémantiques sont basées sur une caractérisation bas-niveau de l'image. Comme nous l'avons dit, cela revient à utiliser l'apparence pour étudier des catégories définies sur des critères fonctionnels. La catégorisation à partir de critères bas-niveau fait donc face à une grande variabilité, ce qui complique le problème.

Une autre approche, inspirée des travaux sur les propriétés perceptuelles des scènes (voir [OT01]), consisterait à chercher quels critères perceptifs sont utilisés pour définir la fonction d'une pièce. Par exemple, un lieu de passage, comme un couloir, est caractérisé par le fait d'être allongé et relativement étroit. À l'opposé, un bureau est un espace plus ouvert et possède des surfaces horizontales pour pouvoir travailler. On voit qu'une telle caractérisation fait appel à des caractéristiques géométriques issues de l'image (et non des caractéristiques géométriques centrées sur le robot comme le donneraient des capteurs lasers). Le champ perceptif du robot étant limité, une telle caractérisation devrait probablement être construite

à partir de l'accumulation temporelle d'images. Un tel programme de recherche pourrait bénéficier de collaborations avec des ergonomes.

La reconnaissance de lieux sémantiques est un problème en apparence très simple mais qui se révèle difficile à réaliser de manière robuste. En particulier, les approches bas-niveau de la catégorisation souffrent toujours du gouffre sémantique. Ainsi, les futurs travaux devront s'intéresser prioritairement à cette tâche.

Pour conclure, nous voyons qu'il reste encore beaucoup de travail dans ce domaine riche, à la croisée de nombreuses disciplines.

Références

[AV07] D. Arthur et S. Vassilvitskii. « k-means++ : The advantages of careful seeding ». Dans : *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial et Applied Mathematics. 2007, p. 1027–1035 (cf. p. 238).

[FD07] B. J. Frey et D. Dueck. « Clustering by Passing Messages Between Data Points ». Dans : *Science* 315 (2007), p. 972–976 (cf. p. 238).

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 235, 236).

[OT01] A. Oliva et A. Torralba. « Modeling the shape of the scene : A holistic representation of the spatial envelope ». Dans : *International Journal of Computer Vision* 42.3 (2001), p. 145–175 (cf. p. 239).

[Pro11] A. Pronobis. « Semantic Mapping with Mobile Robots ». Thèse de doct. Stockholm, Sweden : Royal Institute of Technology (KTH), juin 2011. isbn : ISBN 978-91-7501-039-7 (cf. p. 238).

[RMM10] M. Rudnaya, R. Mattheij et J. Maubach. « Evaluating sharpness functions for automated scanning electron microscopy ». Dans : *Journal of Microscopy* 240.1 (2010), p. 38–49 (cf. p. 237).

[Tir10] P. Tirilly. « Traitement automatique des langues pour l'indexation d'images ». Thèse de doct. Rennes, France : Université de Rennes 1, juil. 2010 (cf. p. 238).

[VS02] J. Vesanto et M. Sulkava. « Distance matrix based clustering of the self-organizing map ». Dans : *Proceedings of the 12th International Conference on Artificial Neural Networks (ICANN 2002)*. 2002, p. 951–956 (cf. p. 238).

[WCR09] J. Wu, H. Christensen et J. Rehg. « Visual place categorization : Problem, dataset, and algorithm ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE. St. Louis, USA, 2009, p. 4763–4770 (cf. p. 236).

A. Bases de données utilisées

Nous présentons les bases de données utilisées dans notre travail et les sources de variabilité qu'elles couvrent (voir section III.3). Nous présentons aussi les plans expérimentaux que nous utilisons durant notre travail.

De manière générale, ces bases ont été acquises par un système d'acquisition guidé manuellement au travers d'un environnement.

Les deux bases de données explorent des environnements différents : COLD utilise des environnements de bureaux et la base VPC utilise des environnements de maisons. Cela nous permet d'explorer une grande diversité de lieux.

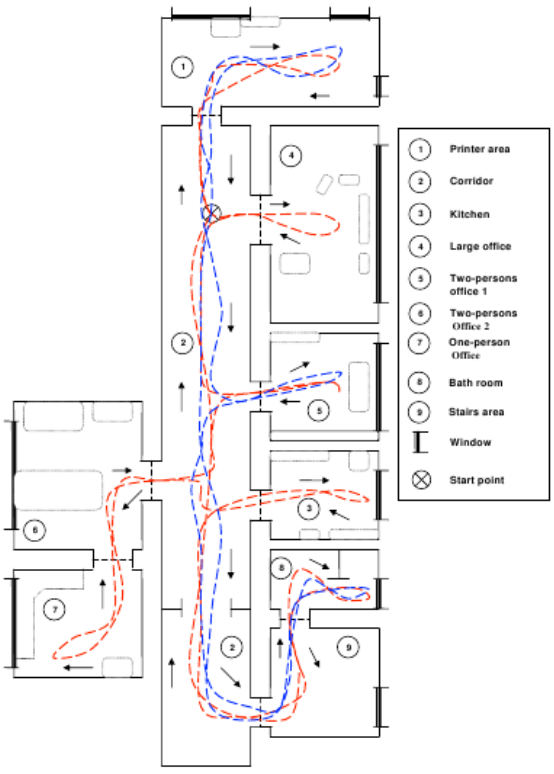
1. Base de données COLD

La base de données *COsy Localization Database (COLD)* (voir [Ull+07 ; PC09] et le site web : `http://cogvis.nada.kth.se/COLD/`) a été acquise dans le cadre du projet européen *Cognitive Systems* (CoSY) spécialement pour la reconnaissance de lieux par un robot mobile dans le but de mieux évaluer ces systèmes. Un grand effort a été fait pour capter différentes sources de variations. Par exemple, comparée, à la base IDOL2 (voir `http://cogvis.nada.kth.se/IDOL2/`), COLD contient beaucoup plus de séquences. De plus, les images proviennent de différents laboratoires, alors que les séquences de IDOL2 proviennent du même laboratoire.

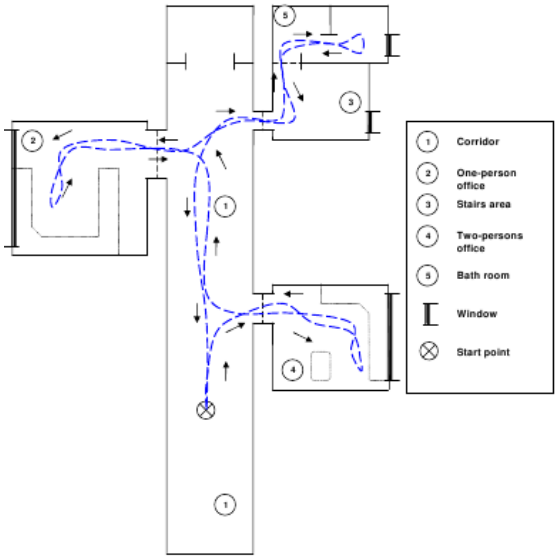
1.1. Environnements

La base de données a été acquise dans trois laboratoires répartis en Europe : le *Visual Cognitive Systems Laboratory* de l'université de Ljubljana en Slovénie, le *Autonomous Intelligent Systems Laboratory* à l'université de Fribourg en Allemagne et le *Language Technology Laboratory* du Deutsches Forschungszentrum für Künstliche Intelligenz (centre allemand de recherche sur l'intelligence artificielle) à Sarrebruck en Allemagne. Dans les laboratoires de Fribourg et Sarrebruck, deux parties différentes ont été utilisées qui sont considérées comme des environnements différents mais visuellement proches. La figure Figure A.1 montre une carte des cinq environnements. On peut donc tester la robustesse à la variabilité environnementale.

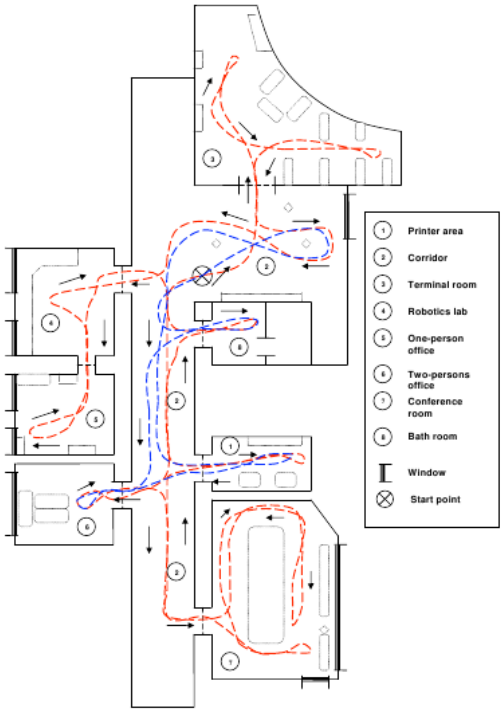
On trouve donc des pièces qui ont la même fonction dans les trois environnements (couloirs, bureaux, reprographie, *etc.*) mais aussi des pièces propres à chaque environnement.



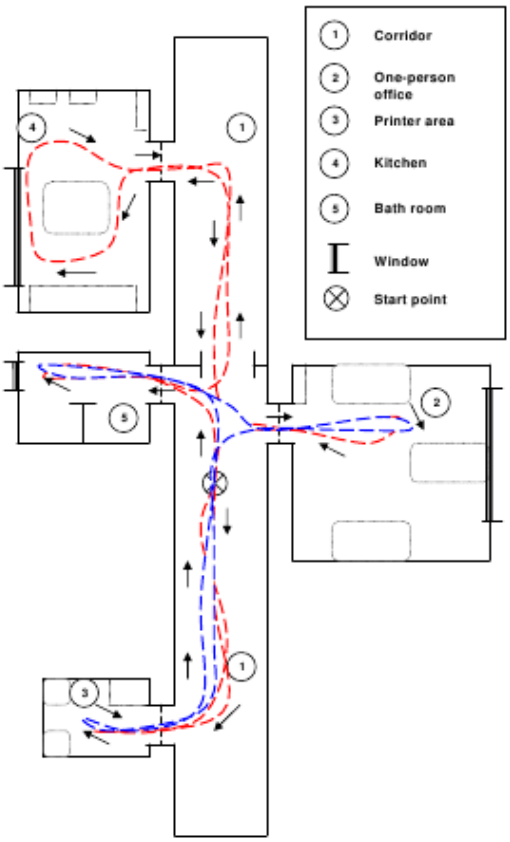
(a) Fribourg, partie A



(b) Fribourg, partie B

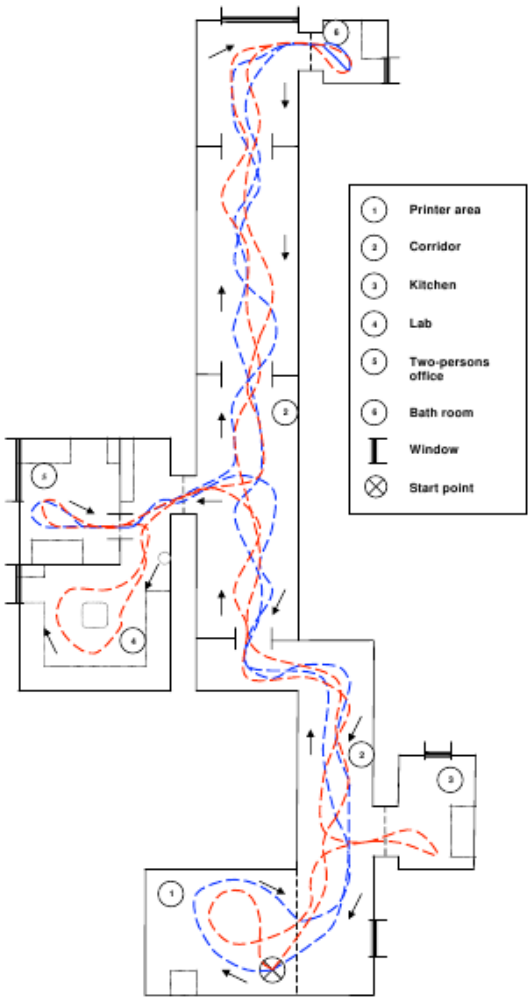


(c) Sarrebruck, partie A



(d) Sarrebruck, partie B

FIGURE A.1.: Cartes des différents environnements utilisés pour la base COLD.



(e) Ljubljana

FIGURE A.1.: (suite) Cartes des différents environnements utilisés pour la base COLD. Les deux chemins suivi dans chaque laboratoire sont montrés. Issu de [Ull+07].

1.2. Procédure d'acquisition

Chaque laboratoire utilise un robot différent. Néanmoins, tous utilisent le même système d'acquisition vidéo composé d'une caméra classique et d'une caméra omnidirectionnelle utilisant un miroir hyperbolique. La hauteur de la caméra varie entre 66 cm et 159 cm. Notre travail utilise uniquement les images de la caméra traditionnelle. Les images ont une résolution de 640×480 pixels et sont acquises avec les corrections automatiques de la caméra.

Le robot est piloté manuellement à une vitesse d'environ 0,3 $^{m}/_{s}$. Les images sont acquises à une fréquence de 5 Hz approximativement ; les deux caméras sont synchronisées. Durant l'acquisition, les personnes ont une activité normale.

Dans chaque laboratoire, deux chemins différents ont été suivis (voir Figure A.1) : un chemin court (appelé standard) et un chemin long (appelé *extended*) qui contient plus de pièces. Le chemin court contient les pièces les plus courantes dans un environnement de bureau.

Les données ont été acquises dans trois conditions de luminosité ambiante : jour ensoleillé, jour nuageux et nuit, ce qui permet d'étudier la robustesse des algorithmes au changement de luminosité. Cependant, certaines conditions n'ont pas pu être testées.

Si possible, il y a eu plusieurs acquisitions de la même séquence (voir Table A.2). Cela permet de tester la robustesse aux changements dynamiques et aux changements de point de vue (le robot étant guidé manuellement, le chemin suivi à chaque répétition n'est pas tout à fait le même).

Il y a au total 32 lieux différents répartis en 12 catégories de lieux (voir Table A.1). On peut noter que certaines catégories pourraient être regroupées en une catégorie plus générale : par exemple, les catégories « bureau 1 personne », « bureau 2 personnes » et « grand bureau » pourraient être regroupées dans une catégorie « bureau ».

Chaque séquence contient entre 4 et 9 catégories. Le nombre d'images par séquences varie entre 646 et 2 942. Il y a au total 137 058 images.

L'étiquetage des données est automatique. La pose du robot dans un repère global, propre à chaque laboratoire, est estimée, ce qui permet de positionner le robot sur les cartes de la Figure A.1. Le lieu courant est estimée par un découpage prédéfini de la carte. Dans les laboratoires de Fribourg et Sarrebruck, l'estimation de la pose fait appel à un algorithme de SLAM. Le robot utilisé à Ljubljana ne dispose pas de lasers. Dans ce laboratoire, la pose est estimée à partir des données odométriques corrigées manuellement (cette procédure contient d'ailleurs quelques erreurs).

Couloir (CR)	Escaliers (ST)[a]	Toilettes (TL)[a]	Cuisine (KT)
Bureau 1 pers. (1PO)	Bureau 2 pers. (2PO)	Grand bureau (LO)	Salle des terminaux (TR)
Salle de conf. (CNR)	Reprographie (PA)	Lab. robot. (RL)	Laboratoire (LAB)

[a] Les notations diffèrent de [Ull+07]. Voir Section A.1.3.

TABLE A.1.: Catégories utilisées dans la base COLD.

	Sarrebruck				Fribourg				Ljubljana	
	A		B		A		B		A	
	Standard	Long	Standard	Long	Standard	Long	Standard	Long	Standard	Long
Nuageux	3	3	5	3	3	3	3		3	3
Nuit	3	3	3	3	3	3			3	3
Ensoleillé			3	3	4	4	3		3	3

TABLE A.2.: Nombre de séquences pour chaque environnement, chemin et condition d'illumination dans la base COLD. Les cases vides indiquent les conditions non-testées. Il y a au total 76 séquences. Issu de [Ull+07].

1.3. Problèmes connus

Lors de notre utilisation, nous avons pu constater quelques erreurs dans la base COLD. Nous avons signalé ces erreurs qui sont recensées à l'adresse : `http://cogvis.nada.kth.se/COLD/bugs.php`.

Ces erreurs sont de trois types :

1. Pièces manquantes dans une séquence : lors de l'acquisition de certaines séquences, les manipulateurs ont oublié de visité une ou plusieurs pièces. Dans nos expériences, nous évitons toujours d'utiliser les séquences entachées de pièces manquantes.
2. Erreurs dans les labels : certains labels reportés dans [Ull+07] sont incorrects : notamment, la catégorie « toilettes » est supposée être étiquetée « BR » (*bathroom*) mais est en fait étiquetée « TL ». Nous utilisons donc cette étiquette.
3. Différents labels utilisés pour la même catégorie : dans certaines séquences, les escaliers sont étiquetés « ST » (*stairs*) et dans d'autres « SA » (*stairs area*). Nous avons choisi d'utiliser « ST ».

1.4. Protocole de reconnaissance d'instances de lieux

ULLAH et al. [Ull+07] présentent un protocole de reconnaissance d'instances sur la base COLD qui permet de tester la robustesse aux variations dynamiques (voir Section III.3.1.1). L'apprentissage est réalisé avec une séquence et l'évaluation se fait pour une séquence issue

de la même partie du même laboratoire mais éventuellement sur une condition d'illumination différente. Le protocole permet donc de tester la robustesse à l'illumination.

De plus, on a plusieurs répétitions de la condition d'entraînement et de la condition de test. Les résultats sont obtenus pour chaque permutation de l'ensemble d'apprentissage et de l'ensemble de test (en s'assurant que ceux-ci sont différents) et moyennés. Les séquences contenant des erreurs (voir Section A.1.3) sont ignorées.

Au total, on va réaliser 74 apprentissages. Chaque classificateur sera évalué entre 4 et 10 fois (au total, on effectue 568 reconnaissances). La Table A.3 donne l'ensemble des conditions qui sont testées.

1.5. Protocole de catégorisation

Ullah et al. [Ull+07] présentent aussi un protocole de catégorisation sur la base COLD. Le protocole utilise quatre catégories qui sont présentes dans tout les laboratoires (on n'utilise que la partie « A ») : « Reprographie », « Couloir », « Bureau 2 personnes » et « Toilettes ».

On apprend avec une séquence tirée dans deux des trois laboratoires pour une condition d'éclairage donnée et on reconnaît pour une séquence tirée dans le troisième laboratoire pour la même condition d'éclairage. Toutes les combinaisons de séquences sont essayées et les résultats sont moyennés entre les répétitions.

Le nombre d'images utilisées pour l'apprentissage varie entre 231 et 516 pour la classe « Reprographie », 1 448 et 3 110 pour la classe « Couloir », 214 et 389 pour la classe « Bureau 2 personnes » et entre 282 et 501 pour la classe « Toilettes ».

À cause des séquences manquantes dans la base, seules les conditions « nuageux » et « nuit » peuvent être testées.

2. Base de données VPC

La base de données Visual Place Categorization (VPC) (voir [WCR09 ; Wu09] et le site web : http://categorizingplaces.com/dataset.html) a été acquise spécifiquement pour étudier le problème de la catégorisation de lieux sémantiques d'intérieur. Un effort particulier a porté sur les variations environnementales. En revanche, la base ne contient pas de variations dynamiques.

2.1. Environnements

La base de données a été acquise dans six maisons avec leur aménagement propre (excepté les objets qui pourraient révéler l'identité ou l'adresse des occupants) mais sans les occupants. Dans chaque maison plusieurs étages ont été utilisés (sauf lorsque cela n'est pas possible).

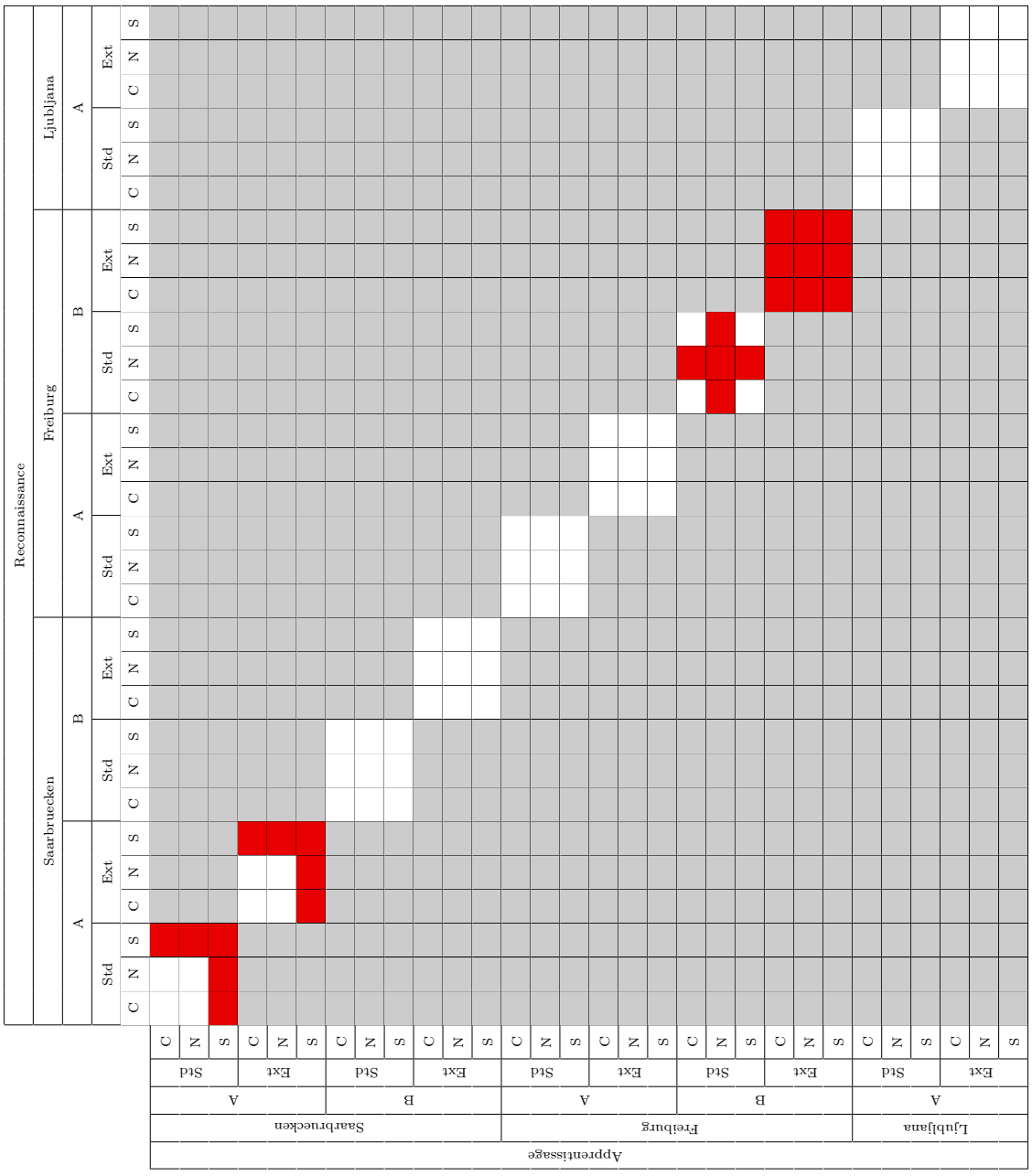


TABLE A.3.: Conditions expérimentales pour le protocole de reconnaissance d'instances. Les cellules en blanc sont les cas testés. Les cellules en gris ne sont pas testées. Les cellules en rouge ne sont pas utilisés car elles correspondent à des conditions non-existantes. S représente la condition « ensoleillée », N la condition « nuit », C la condition « nuageux »

Bedroom	*Bathroom*	*Kitchen*	*Storage closet*
Living room	*Dining room*	*Family room*	*Workspace*
Exercise room	*Media room*	*Corridor*	*Transition*[a]

[a] Catégorie spéciale pour les transitions.

TABLE A.4.: Catégories utilisées dans la base VPC.

Les maisons sont de taille et de style architectural différents (maisons modernes ou plus classiques), incluant des familles avec ou sans enfants.

Les images sont étiquetées uniquement avec la catégorie (ainsi, si deux chambres sont présentes, elles seront toutes les deux étiquetées « Bedroom »). Il y a au total 11 catégories (voir Table A.4) plus une spéciale utilisée pour étiqueter les zones de transition. Notons que, par rapport à la base COLD, les catégories sont plus génériques. Le nombre total d'images est 51 337.

2.2. Procédure d'acquisition

Les images sont acquises à l'aide d'une caméra classique montée sur un tripode roulant à une hauteur d'environ 1 m. Ce système est plus facile à manipuler qu'un robot, surtout dans les zones étroites. Le même système est utilisé dans toutes les maisons.

Les vidéos sont acquises avec les réglages standard et les corrections automatiques de la caméra. Une image sur trois est extraite des vidéos acquises durant l'exploration. La fréquence d'acquisition des images est donc de 10 Hz. Les images ont une résolution de $1\,280 \times 720$ pixels.

En complément, une prise de vue panoramique de chaque pièce a été réalisée en faisant tourner lentement la caméra sur elle-même au centre de la pièce mais ces données ne sont pas disponibles publiquement.

Pour diminuer les effets de l'illumination extérieure, lors de l'acquisition les volets (ou les stores) sont fermés et les lumières sont allumées. Les acquisitions n'ont pas été répétées (on ne peut donc pas tester la robustesse aux variations de point de vue ou aux variations dynamiques).

L'étiquetage des images est réalisé manuellement.

2.3. Protocole de catégorisation

Le protocole proposé par WU [Wu09] utilise cinq catégories présentes dans toutes les maisons (« *bedroom* », « *bathroom* », « *kitchen* », « *living room* » et « *dining room* »). L'apprentissage utilise une cinq maisons et la reconnaissance se fait sur la sixième (cette stratégie est appelée

Classe	Apprentissage	Reconnaissance
« *Bedroom* »	7 859-10 026	823-2 990
« *Bathroom* »	5 287-6 558	663-1 934
« *Kitchen* »	2 508-2 855	223-839
« *Living room* »	2 135-3 005	82-952
« *Dining room* »	2 013-2 457	225-669

TABLE A.5.: Nombre minimum et maximum d'images par classe dans les échantillons d'apprentissage et de reconnaissance pour le protocole Section A.2.3.

leave-one-out). Le nombre d'images d'apprentissage et de reconnaissance est donné dans la Table A.5.

WU [Wu09] évalue le Taux de Vrais Positifs (TVP) par catégorie. L'évaluation finale est la moyenne de ces taux.

3. Stockage des données [1]

Les bases COLD et VPC sont constituées de fichiers répartis selon une certaine organisation sur un disque dur. La taille totale des dossiers est de 47 Go pour COLD et de 10 Go pour VPC. Le stockage de telles masses de données n'est pas sans poser problèmes. De plus, les expériences requièrent de naviguer dans la hiérarchie pour quérir les images nécessaires. Ce genre de processus est laborieux et très sensible aux erreurs. De plus, le calcul des signatures et l'entraînement des cartes de Kohonen sont des étapes longues qui, en général, ne sont faites qu'une fois.

Pour pallier ces problèmes, il nous a parut intéressant d'utiliser une véritable base de données relationnelle afin d'accéder aux données. Nous avons développé une architecture distribuée dans laquelle les images elles-mêmes sont stockées sur un serveur de fichiers, accessible depuis plusieurs machines.

La Figure A.2 montre une partie du schéma de la base de données que nous avons utilisé (par souci de lisibilité, nous avons supprimé quelques tables de signatures et certains attributs des tables). Les tables `image`, `cold_image` et `vpc_image` implémentent une structure d'héritage vertical qui permet de simuler l'héritage tel que pratiqué en programmation objet : la table `image` correspond à la classe abstraite de base et les tables `cold_image` et `cold_image` correspondent aux classes concrètes. Une image est obtenue par jointure entre `image` et la table concernée. La table `signature` joue un rôle similaire de table abstraite pour les différents

1. Les travaux décrits dans cette partie ont été réalisés en grande partie avec nos stagiaires Mohamed Jaite (en 2009) et Christophe Elavumootil (en 2011), avec l'aide inestimable de Laurent Pointal.

types de signatures. Cette table stocke aussi l'association entre une image et une signature. Notons l'existence d'un type particulier de signature dans la table `sig_pca`. Cette table contient la projection d'une signature déjà existante sur un certain nombre de dimensions principales. La base de projection est stockée dans la table `pca` (la table `signature_has_pca` servant à mémoriser quelles signatures ont été utilisées pour calculer la base de projection). Enfin, la table `kohonen_map` sert à stocker une carte de Kohonen dûment entraînée (la table `signature_has_kohonen_map` jouant un rôle analogue à `signature_has_pca`). Il faut noter que notre travail ne respecte pas tout à fait certains principes des bases de données.

Les avantages d'une telle architecture sont nombreux :

1. Plusieurs calculs peuvent être effectués en parallèle : plusieurs hôtes peuvent accéder simultanément aux images et à la base SQL. Cela nous permet par exemple de calculer plusieurs types de signatures en parallèle sur des machines différentes.
2. Une grande souplesse dans les requêtes des images : on peut utiliser le langage SQL pour faire les requêtes de manière souple, sans craindre les erreurs.
3. Les calculs lourds ne sont effectués qu'une fois : une fois ces calculs effectués, les expériences décrites dans les sections A.1.4 à A.2.3 sont effectuées en quelques minutes.
4. La reproductibilité des expériences : toutes les expériences utilisant les mêmes données (signatures, cartes de Kohonen), il est facile de les reproduire.

Signalons toutefois que cette solution peut avoir des inconvénients. En particulier, le stockage des cartes de Kohonen s'est avéré problématique. Premièrement, cela nécessite de garder trace de l'ensemble d'apprentissage utilisé, ce qui nuit parfois à la souplesse expérimentale. Deuxièmement, l'extension de notre base à d'autres types d'algorithmes de quantification vectorielle n'est pas évidente.

Une amélioration possible serait de retirer les algorithmes de quantification vectorielle de la base et d'utiliser les techniques de mappage [2] relationnel-objet.

Références

[PC09] A. Pronobis et B. Caputo. « COLD : COsy Localization Database ». Dans : *The International Journal of Robotics Research* 28.5 (mai 2009) (cf. p. 243).

[Ull+07] M. M. Ullah et al. *The COLD Database*. Rap. tech. TRITA-CSC-CV 2007 :1. Kungliga Tekniska Högskolan, CVAP/CAS, oct. 2007 (cf. p. 243, 245, 247, 248).

2. Ayant longtemps parcouru le continent de l'étymologie tel un Indiana Jones des lettres, notre lecteur est sans doute au courant que le verbe anglais « *to map* » et le verbe français « mapper » ont tout les deux pour origine le terme latin *mappa mundi*. Ainsi « mappage » n'est pas un anglicisme. Source : http://jargonf.org/wiki/mapper et http://www.etymonline.com/index.php?term=map.

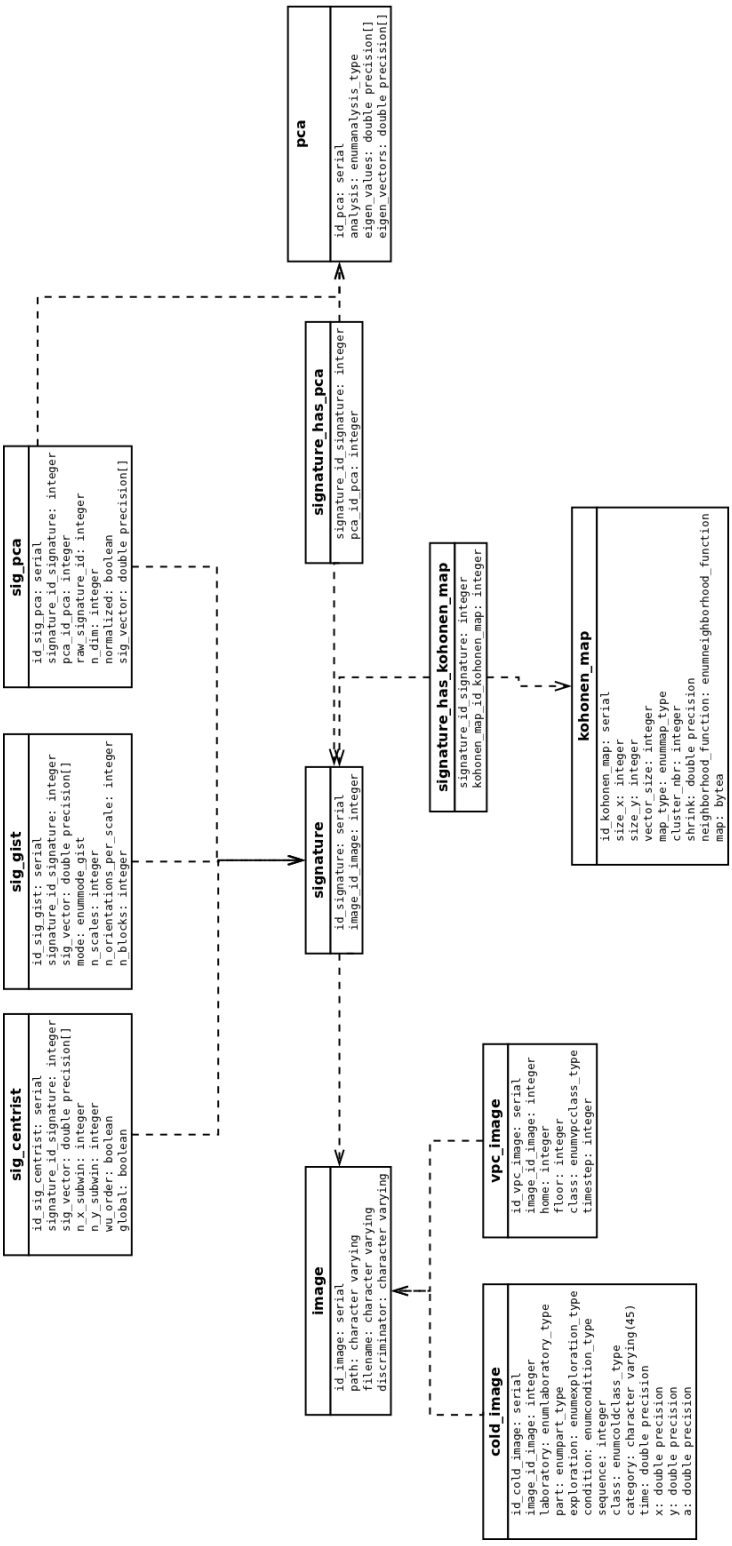


FIGURE A.2.: Schéma de la base de données utilisée dans nos expériences.

[WCR09] J. Wu, H. Christensen et J. Rehg. « Visual place categorization : Problem, dataset, and algorithm ». Dans : *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE. St. Louis, USA, 2009, p. 4763–4770 (cf. p. 248).

[Wu09] J. Wu. « Visual Place Categorization ». Thèse de doct. Georgia Institute of Technology : College of Computing, août 2009 (cf. p. 248, 250, 251).

B. Cartes auto-organisatrices

Nous présentons dans cette annexe les cartes auto-organisatrices que nous avons utilisées comme algorithme de quantification vectorielle. Nous les appellerons indifféremment « cartes auto-organisatrices » ou « cartes de Kohonen ».

1. Principe

1.1. Carte et cellules

Une carte de Kohonen (voir [Koh90]) est un réseau de neurones (appelés cellules ou nœuds) disposés selon une grille de dimension 2 (des extensions en plus grande dimension existent). Chaque neurone représente un centre c'est-à-dire un vecteur $\boldsymbol{\omega} \in \mathcal{X}$, appelé poids ou prototype dans ce contexte. On a S_x cellules horizontalement et S_y verticalement. Il y a donc $K = S_x \times S_y$ cellules. Les neurones sont désignés soit par leurs coordonnées (i, j), soit par un numéro unique k (avec $1 \leq k \leq K$) obtenu par exemple en parcourant les lignes en premier.

La structure en grille permet de définir une relation de voisinage entre les neurones sur la grille. Pour cela, on dispose d'une fonction de distance d entre neurones de la carte.

1.2. Apprentissage compétitif et projection des données

Le but de l'algorithme d'apprentissage est d'adapter les centres de manière à décrire l'ensemble d'apprentissage $\mathcal{S}$. La particularité de l'algorithme de Kohonen est que, une fois la carte entraînée, des cellules proches sur la carte représentent des prototypes proches dans l'espace $\mathcal{X}$ au sens d'une fonction de distance Δ. Cette propriété est appelée auto-organisation (d'où le nom de carte auto-organisatrice). On dit aussi que la carte respecte la topologie du jeu de données (voir figure B.1c).

On commence par initialiser les centres, par exemple avec des valeurs aléatoires ou avec des vecteurs tirés aléatoirement dans l'ensemble d'apprentissage. À cette étape, la topologie n'est pas respectée : des cellules voisines sur la carte peuvent représenter des vecteurs de $\mathcal{X}$ très éloignés.

L'algorithme nécessite d'examiner plusieurs fois l'ensemble d'apprentissage. La présentation d'un exemple est appelée un pas de temps t. On fixe en général un nombre maximal d'itérations

t_{max}, avec $1 \leq t \leq t_{max}$.

L'algorithme fonctionne en deux temps :

1. Lors de la présentation d'un vecteur d'apprentissage $\boldsymbol{x}$, on trouve la cellule k_0 qui est la plus proche de $\boldsymbol{x}$ (voir figure B.1a) :

$$k_0 = \underset{1 \leq k \leq K}{\arg\min} \, \Delta\left(\boldsymbol{x}, \boldsymbol{\omega}_k\right) \tag{B.1}$$

   Cette cellule est appelée « vainqueur » ou *Best-Matching Unit (BMU)*, d'où le nom d'apprentissage « compétitif ».

2. On modifie $\boldsymbol{\omega}_{k_0}$ en direction de $\boldsymbol{x}$ pour minimiser la distance $\Delta\left(\boldsymbol{x}, \boldsymbol{\omega}_k\right)$. L'originalité de la règle d'apprentissage consiste à déplacer aussi les cellules voisines (sur la carte) de k_0 en direction de $\boldsymbol{x}$. Cependant, plus un neurone est éloigné du vainqueur, plus l'amplitude de la mise-à-jour diminue : les neurones éloignés (sur la carte) sont moins modifiés (voir figure B.1b). Ainsi, après mise-à-jour, la topologie est mieux respectée (voir figure B.1b).

Une fois l'apprentissage réalisé, le réseau s'est organisé (voir figure B.1c). On peut l'utiliser pour projeter de nouvelles données et déterminer le prototype le plus proche d'un vecteur $\boldsymbol{x}$. Pour cela, on présente le vecteur $\boldsymbol{x}$ et on calcule la BMU grâce à l'Équation B.1. Le vecteur $\boldsymbol{x}$ est ensuite représenté par $\boldsymbol{\omega}_{k_0}$. En général, on peut même représenter le vecteur par l'indice de la BMU, k_0.

2. Description détaillée

2.1. Algorithme d'apprentissage

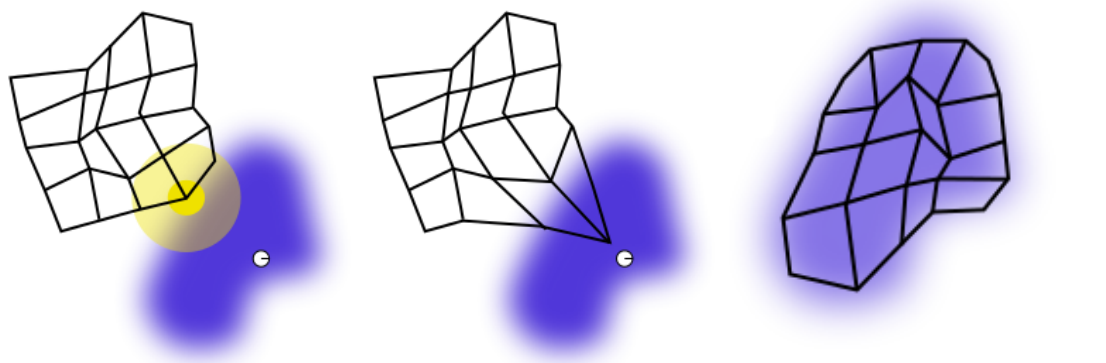
On note $\boldsymbol{x}$ l'exemple courant. Le centre le plus proche de $\boldsymbol{x}$ est noté k_0. Pour minimiser la distance entre un centre et l'exemple courant, l'algorithme effectue une descente de gradient.

De manière générale, la mise à jour des poids de chaque neurone k au temps t s'exprime par la relation :

$$\Delta\omega_k(t) = -\frac{1}{2} \times \mathrm{h}_t\left(\mathrm{d}\left(k, k_0\right)\right) \times \eta(t) \times \frac{\partial}{\partial \boldsymbol{\omega}_k} \Delta\left(\boldsymbol{x}, \boldsymbol{\omega}_k\right)^2 \tag{B.2}$$

où :

- $\eta(t)$ représente le taux d'apprentissage, c'est-à-dire l'amplitude de la variation $\Delta\omega_k(t)$
- $\mathrm{h}_t\left(\mathrm{d}\left(k, k_0\right)\right)$ définit la région d'influence autour du neurone gagnant k_0 : c'est une fonction décroissante de la distance sur la carte entre le neurone k et le neurone gagnant k_0 (on l'appelle parfois noyau de voisinage [Che95]). En général, $\mathrm{h}_t\left(\mathrm{d}\left(k, k_0\right)\right)$ est nul au-delà d'un certain rayon (taille du voisinage) ; cependant, on peut aussi utiliser une fonction tendant asymptotiquement vers 0 (dans ce cas, tous les neurones sont mis à jour).



(a) Sélection du vainqueur (b) Adaptation des poids (c) Après entraînement

FIGURE B.1.: Illustration de l'entraînement d'une Self-Organizing Map (SOM). La distribution des données dans $\mathcal{X}$ est indiquée par le nuage bleu. Le point blanc représente l'exemple courant. (a) : initialement, les centres sont répartis aléatoirement ; la BMU est représentée par le disque jaune et son voisinage est surligné en jaune. (b) : la BMU, ainsi que son voisinage, sont déplacées en direction de l'exemple courant. (c) : à la fin de l'entraînement, les centres sont disposés dans les régions denses de $\mathcal{X}$. Extrait de http://en.wikipedia.org/wiki/Self-organizing_map.

– le gradient $\frac{\partial}{\partial \boldsymbol{\omega}_k} \Delta\left(\boldsymbol{x}, \boldsymbol{\omega}_k\right)^2$ donne la direction de la modification.

Le taux d'apprentissage et la région d'influence sont des fonctions décroissantes du temps t. L'idée est d'employer initialement des valeurs élevées pour chacun de ces paramètres et de les faire décroître en cours d'apprentissage. Dans un premier temps, les valeurs élevées du taux d'apprentissage et du voisinage permettent de « déployer » les neurones là où les données sont concentrées. La décroissance au cours du temps permet d'affiner progressivement la topologie de la carte par rapport à celle des entrées, puis de raffiner le poids des neurones de façon à les centrer sur les nuages denses des entrées.

Dans le cas où Δ est la distance euclidienne, on a la formule :

$$\Delta\omega_k(t) = \mathrm{h}_t\left(\mathrm{d}\left(k, k_0\right)\right) \times \eta(t) \times \left(\boldsymbol{x} - \boldsymbol{\omega}_k\right) \tag{B.3}$$

L'algorithme d'entraînement est présenté dans l'algorithme B.1. Notons qu'il existe une version « par lot » de cet algorithme.

Algorithme B.1: Algorithme d'entraînement des cartes auto-organisatrices.

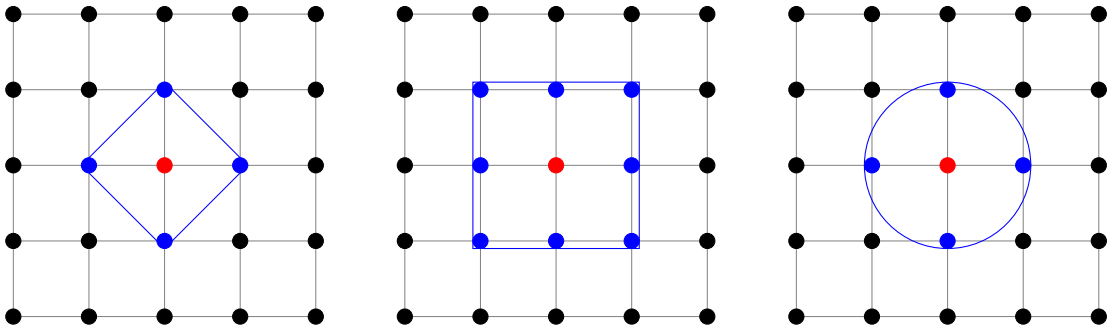
Données : L'ensemble d'apprentissage $\mathcal{S}$
Résultat : Une matrice de prototypes $\{\boldsymbol{\omega}_k, 1 \leq k \leq K\}$

```
Initialiser les centres ω_k;
t = 1;
tant que t ≤ t_max faire // Présentation des exemples
    Tirer au hasard un exemple x ∈ S;
    // Recherche du neurone gagnant;
    k_0 = arg min_{1≤k≤K} Δ(x, ω_k);
    // Mise-à-jour des poids;
    pour 1 ≤ k ≤ K faire
        Δω_k(t) = −(1/2) h_t(k, k_0) η(t) ∂/∂ω_k Δ(x, ω_k)^2;
        ω_k = ω_k + Δω_k(t);
    fin
    t = t + 1;
fin
retourner {ω_k, 1 ≤ k ≤ K}
```

2.2. Voisinage

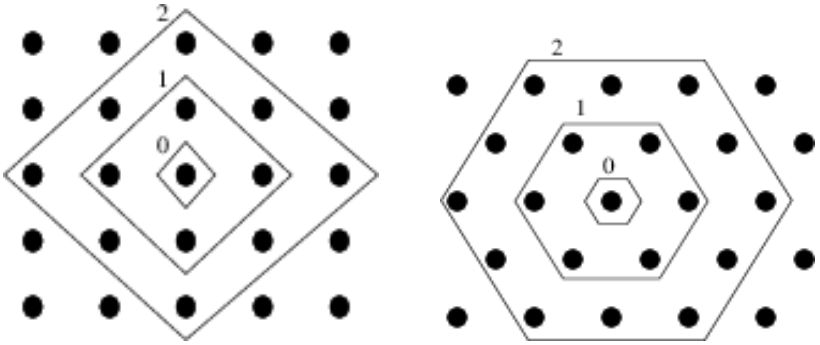
Trois facteurs interviennent dans la définition du voisinage :

1. La fonction de distance utilisée pour calculer la distance sur la carte (voir Figure B.2). Étant donnés deux nœuds $\mathbf{p} = (i, j)$ et $\mathbf{q} = (i', j')$: on peut considérer la distance



(a) Voisinage de Von Neumann. (b) Voisinage de Moore. (c) Voisinage au sens de la distance ℓ_2.

FIGURE B.2.: Différentes fonctions de distance et les voisinages de taille 1 associés.

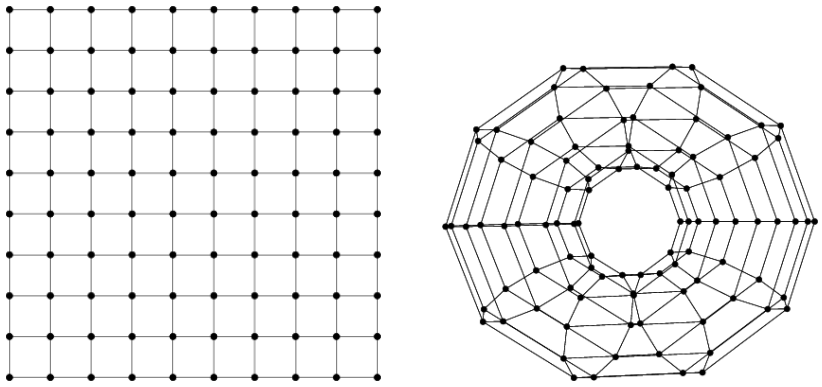


(a) Disposition rectangulaire (b) Disposition hexagonale

FIGURE B.3.: Disposition des cellules dans l'espace 2D. Le schéma montre aussi les voisinages de Moore de taille 0, 1 et 2. Extrait de [Ves+00].

Manhattan $\mathrm{d}(\mathbf{p},\mathbf{q}) = |i-i'| + |j-j'|$ qui définit le voisinage de Von Neumann, la distance de Chebyshev $\mathrm{d}(\mathbf{p},\mathbf{q}) = \max(|i-i'|, |j-j'|)$, qui définit le voisinage de Moore ou encore la distance euclidienne.

2. La disposition des cellules dans l'espace 2D : on peut disposer les cellules à des points de coordonnées entières (voir figure B.3a) ou selon une lattice hexagonale (voir figure B.3b).
3. La forme globale (parfois appelée topologie) de la carte : on peut utiliser des cartes « plates » (voir figure B.4a) ou bien des cartes toroïdales dans lesquelles les bords se touchent (voir figure B.4b). Dans ce dernier cas, le calcul des distances se fait en utilisant le modulo.



(a) Carte plate (b) Carte toroïdale

FIGURE B.4.: Différentes formes de la carte. Extrait de [Ves+00].

3. Paramètres et ensemble d'apprentissage

Les cartes auto-organisatrices et l'algorithme d'entraînement comportent de nombreux paramètres libres. Nous présentons les principaux choix d'implémentation ainsi que les donnés utilisées pour l'entraînement des cartes de Kohonen utilisées dans les chapitres IV, V et VI.

3.1. Paramètres

Nous utilisons en général des cartes carrées ($S_x = S_y = S$), disposées sur une lattice rectangulaire et de forme toroïdale. La distance entre les unités est calculée à l'aide de la distance euclidienne.

Le nombre de pas d'apprentissage est fixé à 5 fois la taille de l'échantillon d'apprentissage. Ainsi en moyenne, chaque exemple est présenté 5 fois.

Les données sont blanchies avant l'entraînement de la carte et avant projection sur la carte. Le blanchiment est une technique de décorrélation qui transforme les données de telle manière que la matrice de covariance est la matrice identité (voir [DHS00, p. 34]). L'initialisation des poids se fait en tirant aléatoirement des vecteurs dans l'ensemble d'apprentissage.

L'algorithme impliquant des étapes aléatoires, il est nécessaire de répéter plusieurs fois l'entraînement. Dans notre cas, nous répétons trois fois la procédure d'entraînement et choisissons la carte qui donne la plus petite distance moyenne entre un prototype et un vecteur de l'ensemble d'apprentissage.

Le taux d'apprentissage évolue en fonction du temps selon une décroissance exponentielle :

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\lambda_\eta}\right) \tag{B.4}$$

avec :

$$\begin{aligned} \eta_0 &= 0.4 \\ \lambda_\eta &= t_{max} \end{aligned}$$

En fin d'apprentissage, $\eta(t_{max}) = \eta_0$.

Le noyau de voisinage est une fonction gaussienne tronquée :

$$\mathrm{h}_t\left(\mathrm{d}\left(k, k_0\right)\right) = \exp\left(-\frac{\mathrm{d}\left(k, k_0\right)}{2\sigma(t)^2}\right) \mathbf{1}\left(\sigma(t) - \mathrm{d}\left(k, k_0\right)\right) \tag{B.5}$$

où $\mathbf{1}$ est la fonction escalier :

$$\mathbf{1}\left(x\right) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

$\sigma(t)$ donne à la fois la taille du voisinage (les neurones à une distance supérieure à $\sigma(t)$ ne sont pas modifiés) et l'importance de la zone d'influence.

Nous faisons également évoluer cette fonction selon une décroissance exponentielle :

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda_\sigma}\right) \tag{B.6}$$

avec :

$$\begin{aligned} \sigma_0 &= \tfrac{S_x + S_y}{2} \\ \lambda_\sigma &= \tfrac{t_{max}}{\log(\sigma_0)} \end{aligned}$$

Ainsi, en fin d'apprentissage, $\sigma(t_{max}) = 1$ (on ne modifie que le neurone gagnant).

3.2. Ensemble d'apprentissage

Les expériences décrites dans l'Appendice A se font soit sur la base COLD, soit sur la base VPC. Ces bases n'ont pas été prises avec le même imageur donc certaines caractéristiques sont différentes (taille, balance de blancs, *etc.*). De plus, les conditions de prises de vues ne sont pas les mêmes (par exemple la luminosité ambiante est variable dans la base COLD mais elle est contrôlée dans la base VPC). Nous avons donc décidé d'entraîner des SOM différentes pour chaque base.

Pour l'entraînement, nous avons utilisé les signatures calculées sur 1/3 des images de la base en question, soit 45 686 signatures pour la base COLD et 17 112 signatures pour la base VPC.

Références

[Che95] Y. Cheng. « Mean shift, mode seeking, and clustering ». Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8 (1995), p. 790–799 (cf. p. 256).

[DHS00] R. O. Duda, P. Hart et D. Stork. *Pattern Classification*. 2e éd. Wiley-Interscience, 2000 (cf. p. 260).

[Koh90] T. Kohonen. « The Self-Organizing Map ». Dans : *Procceding of the IEEE*. T. 78. 9. 1990, p. 1464–1480 (cf. p. 255).

[Ves+00] J. Vesanto et al. *SOM Toolbox for Matlab 5*. Rap. tech. A57. Helsinki University of Technology, 2000 (cf. p. 259, 260).

C. Fonctions de texture

Nous présentons succinctement une partie des fonctions de texture décrites par [HSD73], utilisées par [Gui09]. Les notations sont inspirées de [Pra07, chap. 16].

1. Fonctions de texture sur un histogramme du premier ordre

L'histogramme de premier ordre désigne tout simplement la distribution de probabilité des niveaux d'amplitude du canal. Soit $\mathbf{I}$ une image dont les canaux sont codés sur L niveaux et une fenêtre de taille $H \times W$ centrée autour d'un pixel. L'histogramme du premier ordre autour de ce pixel est donné par :

$$\mathrm{P}(b) = \frac{|\{(x,y), \mathbf{I}(x,y) = b\}|}{H \times W} \tag{C.1}$$

Les caractéristiques sur les histogrammes du premier ordre sont :

1. la moyenne : $\bar{b} = \sum_b b\mathrm{P}(b)$
2. la variance : $\sigma_b^2 = \sum_b (b-\bar{b})^2 \mathrm{P}(b)$
3. l'asymétrie (*skewness*) : $b_S = \frac{1}{\sigma_b^3} \sum_b (b-\bar{b})^3 \mathrm{P}(b)$
4. le kurtosis : $b_K = \frac{1}{\sigma_b^4} \sum_b (b-\bar{b})^4 \mathrm{P}(b) - 3$
5. l'énergie : $b_N = \sum_b \mathrm{P}(b)^2$
6. l'entropie : $b_E = -\sum_b \mathrm{P}(b) \log_2(\mathrm{P}(b))$

2. Fonctions de texture sur un histogramme du second ordre

L'histogramme du second ordre (appelé matrice de co-occurrence dans ce contexte) mesure la densité jointe des pixel entretenant certaines relations géométriques. De même, ces fonctions sont calculées autour de chaque pixel. En pratique, on regarde les couples de pixels décalés

de Δx pixels selon l'axe horizontal et de Δy pixels selon l'axe vertical [1]. L'histogramme du second ordre est alors donné par :

$$\mathrm{P}(a,b) = \frac{|\{(x,y), \mathbf{I}(x,y) = a \text{ et } \mathbf{I}(x+\Delta x, y+\Delta y) = b\}|}{H \times W} \tag{C.2}$$

Les caractéristiques sur les histogrammes du second ordre sont :

1. l'auto-corrélation : $B_A = \sum_a \sum_b ab\mathrm{P}(a,b)$
2. la covariance : $B_C = \sum_a \sum_b (a-\bar{a})(b-\bar{b})\mathrm{P}(a,b)$
3. l'inertie : $B_I = \sum_a \sum_b (a-b)^2\mathrm{P}(a,b)$
4. la valeur absolue : $B_V = \sum_a \sum_b |a-b|\mathrm{P}(a,b)$
5. la différence inverse : $B_D = \sum_a \sum_b \frac{\mathrm{P}(a,b)}{1+(a-b)^2}$
6. l'énergie : $B_N = \sum_a \sum_b \mathrm{P}(a,b)^2$
7. l'entropie : $B_E = \sum_a \sum_b \mathrm{P}(a,b) \log_2(\mathrm{P}(a,b))$

3. Signatures basées sur les fonctions de textures

Pour caractériser les images, nous utilisons soit les histogrammes du premier ordre, soit les histogrammes du second ordre (avec un décalage $\Delta x = \Delta y = 3$), soit les deux. Notons que les fonctions de texture sont calculées sur toute l'image et non autour de chaque pixel. Les meilleurs résultats ont été obtenues avec les signatures calculées sur l'histogramme du second ordre. Nous utiliserons donc uniquement celles-ci.

Les signatures sont calculées :

- soit sur le canal de luminance (on obtient donc un descripteur de taille 7)
- soit sur les canaux de couleurs RGB (on obtient donc un descripteur de taille 21)
- soit sur les canaux de couleurs et le canal de luminance (on obtient donc un descripteur de taille 28).

Références

[Gui09] H. Guillaume. « Explorer pour reconnaître : Approche probabiliste pour la reconnaissance visuelle de lieux par un robot mobile ». Thèse de doct. Université Paris Sud 11 : Départment d'informatique, juil. 2009 (cf. p. 263).

1. On peut aussi utiliser les coordonnées polaires pour cela. Le décalage sera alors donnée par une distance r et un angle θ par rapport à l'axe horizontal.

[HSD73] R. M. Haralick, K. Shanmugam et I. Dinstein. « Textural Features for Image Classification ». Dans : *IEEE Transactions on Systems, Man and Cybernetics* 3.6 (nov. 1973), p. 610–621 (cf. p. 263).

[Pra07] W. K. Pratt. *Digital Image Processing*. 4th. Wiley-Interscience, 2007 (cf. p. 263).

D. Théorie de la détection

La mesure la plus simple de la performance d'un classificateur est bien sûr le taux de classification correcte, c'est-à-dire le taux d'exemples dont la classe prédite correspond à la classe réelle. Cependant, il est parfois nécessaire d'être plus fin dans l'évaluation. En effet, dans certains cas, se tromper de classe peut avoir des conséquences dramatiques. Par exemple, il est plus grave de laisser passer une tumeur (qui peut évoluer en maladie grave) que d'opérer pour rien. Dans un tel contexte, il n'est pas très grave de classer un organe sain comme malade mais il est très grave de classer un organe malade comme sain.

Un autre problème peut se produire dans le cas où une classe est beaucoup plus représentée qu'une autre. Par exemple, dans la base COLD, il est fréquent que le nombre d'images du couloir soit 5 à 6 fois supérieur aux nombre d'images des autres pièces. Supposons que nous ayons 80% d'exemples appartenant à la classe « couloir ». Alors, un système qui répondrait tout le temps « couloir » aurait un taux de classification correcte de 80%. On se doute qu'un tel système n'est pas très intéressant. La théorie de la détection que nous présentons ici donne un cadre rigoureux pour évaluer un système de classification [1].

1. Cas de la classification binaire

1.1. Table de contingence

Supposons que nous nous intéressions à un problème de classification binaire. Dans ce cas, chaque exemple va être classé dans l'une des deux classes. En général, on cherche à distinguer une classe d'intérêt, notée +, du bruit, noté -. Par exemple dans le cas d'un radar, on cherche à savoir si un écho reçu correspond à un appareil (classe d'intérêt) ou si il est causé par l'atmosphère, une défaillance du système, *etc.*. Le système d'apprentissage va donc classer chaque exemple comme appartenant à la classe + ou à la classe -.

En apprentissage supervisé, on dispose de la vérité terrain [2], c'est-à-dire de la vraie classe de

1. Cette théorie a été développée dès la seconde guerre mondiale pour les premiers systèmes de radar, d'où le nom de théorie de la détection. En effet, en temps de guerre, il est crucial de ne pas laisser échapper un appareil ennemi quitte à déclencher quelques fausses alertes. Cette théorie a aussi beaucoup été employée en psychophysique. Étant une science anglaise, elle est dotée d'un vocabulaire folklorique et imagé.

2. Là encore, la référence militaire est évidente. Le lecteur qui voudrait un temps laisser de côté l'étymologie pour batifoler dans la linguistique synchrone, notera que, dans la langue anglaise, on parle maintenant de *ground truthing*, du néologissime verbe *to ground truth*. *O tempora, O mores.*

		Vérité terrain		
		+	-	**Total**
Réponse	+	Vrais positifs (VP)	Faux positifs (FP)	P'
	-	Faux négatifs (FN)	Vrais négatifs (VN)	N'
Total		P	N	

TABLE D.1.: Table de contingence et définitions associées. Les cellules grisées sont les cas où le classificateur répond correctement.

l'exemple. Quatre cas de figures sont possibles selon la conformité des résultats du système de classification à la vérité terrain. Ces résultats sont représentés dans une table de contingence, qui compte le nombre de d'exemple tombant dans chaque cas. La Table D.1 présente un exemple de table de contingence et le vocabulaire décrivant les cas :

- Les vrais positifs correspondent aux exemples positifs classés positivement (la classification est donc correcte). De même, les vrais négatifs correspondent aux exemples négatifs classés correctement.
- Les faux négatifs correspondent aux exemples classés négativement par erreur (ils sont donc positifs). Cela correspond au cas où on n'a pas détecté un avion ennemi. Ce cas est aussi appelé « non-détection » (ou *miss*) ou « erreur de type 2 ».
- Les faux positifs correspondent aux exemples classés positivement par erreur (ils sont donc négatifs). Cela correspond au cas où on a détecté un avion ennemi alors qu'il n'y en a pas. Ce cas est aussi appelé « fausse alarme » ou « erreur de type 1 »

Le nombre d'exemples positifs est donné par : $P = VP + FN$.
Le nombre d'exemple négatifs est donné par : $N = FP + VN$.
On peut alors définir plusieurs métriques :

1. Le Taux de Vrais Positifs (TVP) ou sensibilité ou taux de rappel (*True Positive Rate (TPR)* ou *recall* en anglais) mesure le taux d'exemples positifs classés comme tels :

$$TVP = \frac{VP}{P} = \frac{VP}{VP + FN} \tag{D.1}$$

   Une sensibilité proche de 1 signifie que le système reconnaît bien la classe d'intérêt.

2. Le Taux de Vrais Négatifs (TVN) ou spécificité mesure le taux d'exemples négatifs correctement classés :

$$TVN = \frac{VN}{N} = \frac{VN}{FP + VN} \tag{D.2}$$

Une spécificité proche de 1 signifie que le système reconnaît le fond correctement (il est donc spécifique de la classe d'intérêt).

Notons que si l'on inverse les classes d'intérêt et le bruit, le TVP devient le TVN.

3. Le Taux de Faux Positifs (TFP) ou taux de fausse alarme mesure le taux d'exemples négatifs qui sont classés comme positifs :

$$TFP = \frac{FP}{N} = \frac{FP}{FP + VN} = 1 - TVN \tag{D.3}$$

4. L'exactitude (*accuracy*) est la proportion de résultats justes :

$$E = \frac{VP + VN}{P + N} = \frac{VP + VN}{VP + VN + FP + FN} \tag{D.4}$$

5. La précision correspond au taux d'exemples classés comme positifs qui le sont réellement :

$$P = \frac{VP}{P'} = \frac{VP}{VP + FP} \tag{D.5}$$

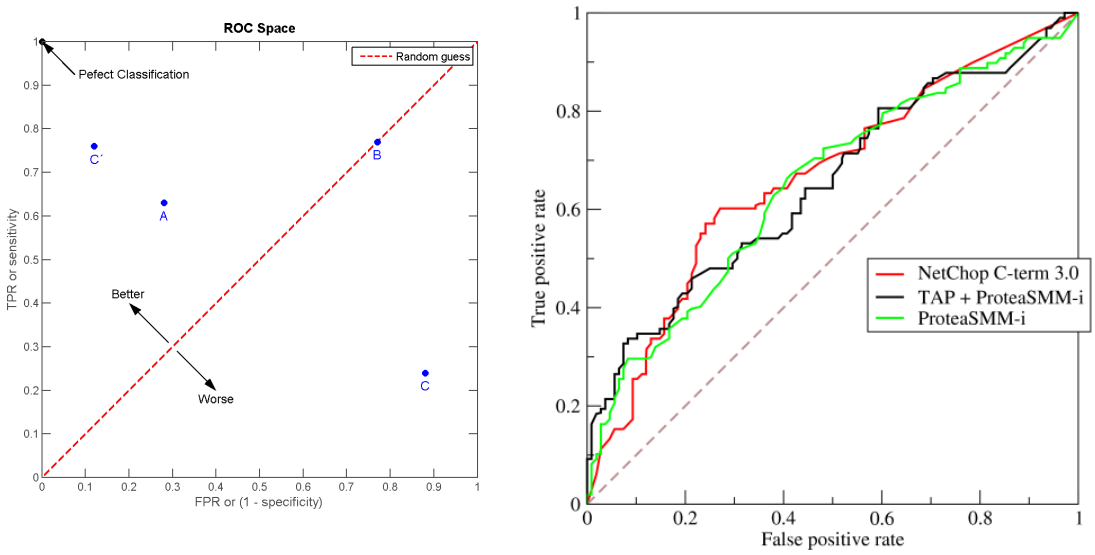
Une valeur proche de 1 signifie que tous les exemples classés comme positifs le sont vraiment.

1.2. L'espace ROC

En général, on souhaite obtenir un système ayant une sensibilité la plus forte possible tout en ne produisant pas de fausses alarmes (donc ayant une forte spécificité). Le compromis entre ces grandeurs est visualisé dans l'espace *Receiver Operating Characteristic (ROC)* dont l'axe horizontal est le TFP et l'axe vertical le TVP (voir figure D.1a). Cet espace est parfois appelé « sensibilité *versus* 1-spécificité ». Dans cet espace, un classificateur est représenté par un point, par exemple le point A dans la figure D.1a.

Si le classificateur répond au hasard, en moyenne, le TFP et le TVP seront égaux et le point représentatif sera sur la ligne marquée en pointillés rouges, comme le point B dans la figure D.1a. Un classificateur parfait aurait une sensibilité égale à 1 et une spécificité égale à 1 (donc un TFP égal à 0). Il serait donc situé dans le coin supérieur gauche de coordonnées $(0, 1)$. Un classificateur situé dans le coin inférieur droit répondrait systématiquement faux. Globalement, plus un classificateur s'approche du coin supérieur gauche, meilleur il est.

Un classificateur situé dans le coin inférieur droit, par exemple le point C dans la figure D.1a, peut être transformé en un bon classificateur en inversant sa sortie. Cela revient à prendre le symétrique par rapport au point central, ce qui donne le point C' dans la figure D.1a.



(a) (b)

FIGURE D.1.: (a) : L'espace ROC. (b) : Exemple de courbes ROC correspondant à trois classificateurs binaires. Extrait de http://en.wikipedia.org/wiki/Receiver_operating_characteristic.

1.3. Courbe ROC

La plupart des classificateurs ne donnent pas directement une classification binaire mais attribuent un score à la classe d'intérêt. La classification est alors obtenue en imposant un seuil sur ce score : si le score de la classe + est supérieur au seuil, l'exemple est classé +.

C'est par exemple le cas avec un classificateur probabiliste qui retourne la probabilité $\mathrm{P}(+|\boldsymbol{x})$. Le principe du maximum *a posteriori* commande de choisir la classe qui a la probabilité *a posteriori* la plus forte. On peut cependant choisir de classer un exemple comme appartenant à + uniquement si $\mathrm{P}(+|\boldsymbol{x})$ dépasse un seuil θ. D'autres types de classificateurs fournissent un score qui n'est pas une probabilité.

Par exemple, si le coût d'une classification est très élevé, on choisira un seuil θ grand (le TFP est alors faible car le système ne classe un exemple que si sa probabilité est très élevée). *A contrario*, si il est plus coûteux de laisser passer un événement, on choisira un seuil θ petit (on risque alors de commettre de nombreux faux positifs). Le seuil θ influe donc sur le compromis entre le TFP et le TVP, c'est-à-dire sur la position du classificateur dans l'espace ROC : un seuil sévère tend à diminuer le TFP et donc à diminuer le TVP (car ils produisent beaucoup de faux négatifs) ; un seuil bas tend à augmenter le TFP et à augmenter le TVP (car ils classent beaucoup d'exemples comme appartenant à +).

La représentation du classificateur pour différentes valeurs du seuil est appelée « courbe

Réponse		Vérité terrain		
		Bureau	**Couloir**	**Entrée**
	Bureau	18	9	0
	Couloir	10	12	9
	Entrée	0	6	36

(a)

Réponse		Vérité terrain	
		+	-
	+	18	9
	-	10	63

(b)

TABLE D.2.: Problème multiclasse : (a) matrice de confusion dans le cas d'un problème à trois classes. Les cellules grisées sont les cas où le classificateur répond correctement. (b) table de contingence associée à la classe couloir.

ROC ». Typiquement, cette courbe a une allure concave (voir figure D.1b). Plus la courbe est incurvée vers le point de coordonnées $(0, 1)$, meilleur est le classificateur.

Les courbes ROC permettent donc de comparer deux classificateurs : si la courbe ROC d'un classificateur est toujours au dessus de celle d'un autre, le premier est meilleur. En pratique, c'est rarement le cas (par exemple sur la figure D.1b). On utilise alors l'aire sous la courbe qui est reliée au test des rangs de Wilcoxon : le classificateur qui a la plus grande aire est le meilleur.

2. Cas d'un problème à plus de deux classes

Dans le cas où l'on a plusieurs classes, on présente les résultats sous forme d'une matrice de confusion qui généralise la table de contingence (voir table D.2a).

L'extension des courbes ROC au cas multi-classes est encore un sujet de recherche. En effet, dans le cas où on a p classes, il faut trouver un compromis entre p bonnes classifications (sur la diagonale) et les $p^2 - p$ erreurs.

On peut obtenir les mêmes métriques que dans le cas binaire en considérant chaque classe tour-à-tour comme la classe d'intérêt. Par exemple, en considérant la classe « bureau » *versus* les autres, on obtient la table de contingence D.2b. On peut ainsi tracer p courbes ROC.