



Un atelier d'ingénierie des connaissances en traitement d'images

Régis Clouard, Arnaud Renouf, Marinette Revenu

► **To cite this version:**

Régis Clouard, Arnaud Renouf, Marinette Revenu. Un atelier d'ingénierie des connaissances en traitement d'images. 8èmes journées francophones extraction et gestion des connaissances, Jan 2008, Sophia Antipolis, France. pp.7-18, 2008. <hal-00256308>

HAL Id: hal-00256308

<https://hal.archives-ouvertes.fr/hal-00256308>

Submitted on 15 Feb 2008

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.

Un atelier d'ingénierie des connaissances en traitement d'images

Régis Clouard, Arnaud Renouf, Marinette Revenu

GREYC, ENSICAEN & Université de Caen
6, boulevard Maréchal Juin, 104050 Caen cedex
Regis.Clouard@greyc.ensicaen.fr
<http://www.greyc.ensicaen.fr>

Résumé. Notre objectif de recherche à long terme est de développer un système à base de connaissances capable de produire automatiquement des logiciels de traitement d'images à partir d'une spécification faite par un utilisateur. Cet objectif a déjà été largement étudié et s'est heurté à un échec. Néanmoins, après avoir analysé les causes de cet échec, nous proposons de reprendre ces recherches en changeant de perspective pour adopter une approche constructiviste de la réalisation du système. Pour cela, il nous est nécessaire de construire un atelier d'ingénierie des connaissances en traitement d'images qui se veut le lieu de l'étude et du développement de ce logiciel. L'atelier se présente comme une ensemble d'outils logiciels partageant une même ontologie destinés à faire collaborer l'expert de traitement d'images et l'ingénieur de la connaissance afin d'aboutir à une théorie du développement d'application opérationnalisable dans un système.

1 Introduction

En dépit du nombre d'applications de traitement d'images réalisées grandeur nature variant autant les objectifs que les classes d'images, le développement d'une nouvelle application est encore décrit comme une tâche particulièrement complexe à la fois laborieuse et surtout à forte charge cognitive. Ceci fait que le développement d'applications de traitement d'images reste l'apanage de spécialistes et freine une réelle pénétration dans le milieu industriel alors que les besoins croissent très fortement à la mesure de l'essor des capteurs d'images.

C'est pourquoi très tôt, les chercheurs se sont intéressés à la conception de systèmes capables d'analyser automatiquement le contenu des images. Ces recherches ont débuté avec l'ambition de la généralité et du tout automatique. Les systèmes les plus emblématiques de ces travaux sont les systèmes d'interprétation d'images VISIONS/SCHEMA (Hanson, 1978) et SIGMA (Matsuyama, 1990). Mais très rapidement, le talon d'Achille de ces systèmes s'est révélé être la partie bas niveau, c'est à dire le traitement des images. L'effort de recherche avait été mis sur la partie haut niveau estimant que les erreurs et insuffisances de la partie bas niveau, réduite à de simples algorithmes généraux de segmentation, pouvaient être compensées par la partie haut niveau grâce aux connaissances modélisées sur le domaine d'application. Mais, l'analyse de ces systèmes montre au contraire que les impérities de la partie traitement d'images sont rédhibitoires pour le fonctionnement du système global (Draper, 1996). Dans un deuxième temps, les recherches se sont alors recentrées sur la partie bas niveau pour

Un atelier d'ingénierie des connaissances en traitement d'images

tenter de la rendre plus adaptative en intégrant dès ce niveau les connaissances sur le domaine d'application. Pour réaliser cela, pratiquement toutes les architectures logicielles de résolution de problème ont été essayées (Crevier, 1997), (Clouard, 1999) :

- compétition d'algorithmes avec recherche de la meilleure solution ;
- systèmes experts ;
- réseaux de neurones ;
- modèles statistiques (réseaux bayésien, processus de décision markovien) ;
- systèmes à base de connaissances.

Malgré cette profusion de travaux qui a atteint son zénith au milieu des années 90, aucun système produit n'a permis une utilisation en routine. Cet échec a donné un violent coup d'arrêt aux travaux sur le développement de système automatique de traitement d'images à vocation générale.

1.1 Autopsie d'un échec

Les raisons de cet échec sont à rechercher d'une part dans la problématique même du traitement d'images et d'autre part dans l'approche de l'acquisition des connaissances injectées dans le système.

1.2 Les raisons liées au domaine du traitement des images

Le traitement d'images se caractérise principalement par la difficulté à poser le problème à résoudre. L'analyse des besoins est typiquement une activité de nature qualitative qui implique qu'il ne peut exister de spécification exhaustive ou exacte de l'application mais simplement une caractérisation approchée du comportement souhaité (Clouard, 2004).

La principale raison tient au fait que le problème à traiter n'est pas dans les images qui sont les seules informations objectives disponibles. En effet, d'une part les données image sont intrinsèquement incomplètes, dégradées et corrompues. La production d'image est responsable de la dénaturation de la scène (*i.e.*, le phénomène observé ou calculé) que ce soit la perte d'information réelle (*e.g.*, 3D, mouvement), le mélange de plusieurs sources dans la valeur d'un pixel dont la contribution relative est impossible à retrouver (*e.g.*, texture, illumination, géométrie), l'introduction de valeurs erronées (*e.g.*, bruit, aberration chromatique) ou la dégradation de l'information originelle (*e.g.*, distortion géométrique, flou). D'autre part, le contenu de l'image n'a pas de sens en lui-même. Une image est ambiguë par nature et ne fournit pas d'information sur son contenu. Sans sujet de traitement, une image ne permet pas de distinguer les informations à considérer comme pertinentes de celles qui ne le sont pas. Par exemple, une information aussi simple que le contour d'objet est difficile à extraire avec précision sans connaissance sur le contenu de la scène. Les contours sont généralement modélisés comme une transition franche d'intensité, mais c'est aussi le cas du bruit, des ombres, de la texture, etc.

Pour espérer résoudre de tels problèmes, dits mal définis, il est nécessaire de disposer d'information sur le contexte de traitement pour contraindre l'espace des solutions. Ces informations ont pour but d'aider à corriger les données dégradées ou corrompues, à séparer les sources composant la valeur d'un pixel et à donner un sujet et un sens aux images. Elles sont disponibles pour une partie sous forme numérique et pour une autre sous forme symbolique. En conséquence pour réaliser un système de traitement d'images, il n'y a que deux solutions :

- faire un système dédié dans lequel les informations sur le domaine d'application sont implicitement codées ;

- faire un système général qui soit capable de collecter et d'organiser les informations numériques et symboliques sur le domaine d'application en connaissances utilisables pour la résolution.

1.3 Les raisons liées à l'acquisition des connaissances

Si on envisage la généralité, le choix de réalisation doit s'orienter vers un système à base de connaissances. En effet, si les traitements relèvent bien d'une approche numérique, le développement d'une solution logicielle complète ne se réduit pas à un problème calculatoire (Garbay-02) (Clouard-2004). La nature qualitative de l'analyse des besoins conditionne toute la chaîne de résolution de problèmes de traitement d'images. En particulier, la résolution ne peut pas s'envisager comme la recherche d'une solution optimale mais simplement comme la recherche d'une solution "satisfaisante" avec toute la subjectivité que cela comporte. Ceci fait du développement d'application de traitement d'images un domaine où l'approche algorithmique, type minimisation de fonctionnelle, apprentissage à partir d'exemples, modélisation statistique, est inapplicable pour concevoir des systèmes non dédiés.

Si les systèmes à base de connaissances sont légitimes, pourquoi ceux-ci n'ont-ils pas permis des systèmes opérationnels ? On peut constater à partir de la variété des approches proposées que ce n'est pas un problème d'architecture. La cause de l'échec est en fait à rechercher dans le goulot d'étranglement que représente l'acquisition des connaissances.

La première erreur, c'est d'avoir cherché à construire des modèles de connaissances qui soient le reflet du savoir des experts consultés, à partir d'une verbalisation de leurs connaissances et de leur démarche de résolution (Bachimont, 1996). Or, les connaissances de traitement d'image sont tacites et peu verbalisables et la démarche de résolution est essentiellement empirique en l'absence d'une théorie formelle du domaine. L'hypothèse de l'existence d'une expertise rationnelle s'est donc révélée fautive et la tentative d'acquisition des connaissances s'est avérée vaine.

La deuxième erreur, c'est d'avoir cru à l'hypothèse du monde fermé impliqué par la modélisation des connaissances a priori (Draper, 1996). L'essentiel de l'effort de modélisation s'est porté sur la représentation de différentes expertises de résolution de problèmes qui ont pu être identifiées dans le domaine du traitement d'images parce que l'on a fait l'hypothèse que le domaine se caractérisait par ses solutions. La formulation du problème se réduit alors à choisir une solution dont l'étiquette est libellée par le nom d'une tâche. L'utilisateur a l'impression de poser le problème mais en fait il choisit une solution. De plus pour rendre ces solutions effectivement opérationnelles, bon nombre de connaissances a priori y ont été implicitement codées, ce qui les a rendu peu réutilisables. Les informations sur le contexte (*e.g.*, caractéristiques des capteurs, des régions, des contours) servent simplement à paramétrer cette solution. Les systèmes produits se sont donc révélés peu adaptables parce que les solutions n'étaient pas construites spécifiquement pour le problème traité mais prédéfinies et simplement ajustées avec les informations du contexte.

1.4 Notre solution

Notre objectif de recherche à long terme consiste à concevoir un système à base de connaissances capable de produire automatiquement des logiciels de traitement d'images pour tout type d'application. Pour ne pas aboutir au même échec, nous avons choisi d'adopter une approche constructiviste de la construction du système à base de connaissances (Charlet, 2002). Il s'agit d'un changement de perspective pour passer d'une modélisation de l'expert et de son fonctionnement cognitif à une modélisation du système et de son fonctionnement ef-

fectif (Gaines, 1993).

C'est dans cette perspective que s'inscrit notre atelier d'ingénierie des connaissances. Il fournit l'environnement logiciel qui doit permettre d'élaborer les modèles originaux du développement d'application de traitement d'images qui seront ensuite opérationnalisés dans un système à base de connaissances.

La section 2 présente les objectifs, les hypothèses de base et l'architecture de notre atelier logiciel dans lequel les ontologies jouent un rôle central. La section 3 détaille la partie de l'atelier qui concerne la formulation d'application de traitement d'images. Nous l'avons souligné plus tôt, la formulation est un point critique dans la conception d'un système à base de connaissance pour le traitement d'images. En conclusion, nous faisons un bilan de l'état actuel de l'atelier puis des perspectives à court terme.

2 Un atelier d'ingénierie des connaissances

L'atelier a pour raison d'être la conception d'un système de génération automatique de logiciels de traitement d'images. Un tel système doit permettre à l'utilisateur de construire lui-même son logiciel dédié par interaction avec l'interface. L'utilisateur est considéré comme naïf du traitement d'images mais spécialiste d'un domaine d'application, c'est à dire qu'il a une réelle expertise dans le domaine des images et qu'il a un projet. Le logiciel produit est supposé capable de traiter toutes les images de la classe identifiée.

L'atelier se propose d'analyser toute la chaîne de développement depuis la formalisation des problèmes, la construction d'une solution logicielle jusqu'à son évaluation dans le but de construire des modèles qui forment les éléments d'une théorie du domaine du développement d'applications de traitement d'images du point de vue du système devant traiter ce problème. Pour cela, l'atelier doit être le lieu de l'étude de la connaissance en contexte. C'est l'interaction entre l'expert du traitement d'images et l'ingénieur de la connaissance par l'intermédiaire des outils logiciels fournis par l'atelier qui crée la connaissance (Linster, 1993).

L'atelier couvre tout le domaine du traitement d'images, c'est à dire qu'il est indépendant de tout domaine métier mais qu'il prend en compte toutes les tâches possibles en traitement d'images.

2.1 Le traitement d'images comme contexte

En l'absence de définition consensuelle du domaine du traitement d'images, nous proposons une définition établie selon un point de vue intentionnel. Le traitement d'images se définit à partir des six catégories de tâches atomiques résumées dans le tableau 1. Ces six catégories résultent de la combinaison des trois opérations de base - ajout, modification et suppression - appliquées à des fins de pure transformation ou à des fins épistémiques (augmentation de l'intelligibilité de l'information).

- La *restauration* a pour but de retrouver l'image d'origine à partir de sa version dégradée par ajout d'information au niveau signal connaissant la fonction de dégradation (ou au moins une estimation).
- L'*amélioration* a pour but d'adapter au mieux les données à une exploitation visuelle par modification de l'information au niveau signal en utilisant uniquement les valeurs de l'image d'entrée.
- La *compression* a pour but de réduire la quantité de données nécessaire pour stocker une image numérique en supprimant l'information identifiée comme redondante au niveau signal.

- La *reconstruction* consiste à créer une nouvelle image par ajout d'une information spatio-temporelle déduite d'une image ou de plusieurs (*e.g.*, la forme, le relief, le mouvement).
- La *segmentation* structure l'information signal à un niveau d'information symbolique : points d'intérêt, régions et contours.
- La *détection* fournit un masque positif des zones de l'image contenant les objets d'intérêt et supprime toutes les autres informations.

	Transformation	Augmentation
Ajout	<i>Restauration</i>	<i>Reconstruction</i>
Modification	<i>Amélioration</i>	<i>Segmentation</i>
Suppression	<i>Compression</i>	<i>Détection</i>

TAB. 1 – *Le traitement d'images recouvre six catégories de tâches.*

Une application complète de traitement d'images nécessite en généralement l'enchaînement de plusieurs de ces tâches élémentaires où une sortie considérée au niveau symbole peut se retrouver comme entrée au niveau signal dans la tâche suivante. Par exemple, une application de numérisation de livre ancien enchaîne une étape de reconstruction pour une remise à plat de la page par "shape from X" puis une amélioration du rendu visuel en homogénéisant la surface de chaque page avant la compression pour l'archivage numérique.

Cette hypothèse de l'existence d'un domaine du traitement d'images ne doit pas faire croire que nous allons retomber dans l'écueil du monde fermé. Notre thèse, c'est de considérer qu'il existe bien un domaine du traitement d'images indépendant des domaines d'application. Les connaissances de résolution ne se définissent pas par rapport à un domaine d'application mais bien par rapport au domaine de l'image numérique. Pour schématiser, il n'y a pas de technique de traitement d'images propre aux domaines biomédical ou astronomique ; ces techniques sont réutilisables en l'état dans d'autres domaines. Par contre, les informations sur le domaine d'application traité sont nécessaires à l'élaboration d'une solution. Il est donc indispensable d'acquérir ces informations auprès de l'utilisateur pour chaque application. Cela conduit à faire un vrai travail de modélisation de la formulation d'application qui doit distinguer ce qui doit être représenté dans le système de ce qui doit être acquis pour une application particulière.

2.2 Les hypothèses de base

L'hypothèse forte à la base de notre atelier, c'est de considérer que les connaissances de traitement d'images ne sont pas disponibles sous forme linguistique. Des travaux comme ceux de l'équipe LITIS de Rouen ont fait l'hypothèse inverse et ont basé l'élaboration de leur systèmes sur l'analyse de corpus linguistiques (interview, articles scientifiques et thèse) (Saidali-07) (Labiche-07). Notre point de vue, c'est de considérer que les corpus linguistiques (littérature ou interview) ne rendent pas compte d'une théorie de la résolution. S'ils détaillent bien les techniques du traitement d'images c'est uniquement par ce qu'elles sont, un peu par ce qu'elles font mais pas du tout par leurs conditions d'application. La raison est encore une fois que ces connaissances sont tacites et peu verbalisables. Ceci fait du traitement d'images un domaine très particulier pour lequel beaucoup de modèles et de méthodologies d'ingénierie des connaissances ne sont pas utilisables.

A défaut de corpus linguistique constitué, il nous est nécessaire de construire nos propres

modèles de capitalisation des connaissances afin de constituer les corpus. Ce qui est immédiatement disponible en traitement d'images, c'est le code du programme qui constitue l'application. Toutefois, nous proposons de capitaliser non pas les programmes directement mais une représentation sous la forme de graphes d'opérateurs. La notion de graphe d'opérateurs individualisés pour représenter les programmes de traitement d'images est maintenant communément admise dans la communauté du domaine du traitement d'images (Draper-1999). Il est seulement nécessaire d'ajouter dans la représentation des graphes tous les éléments d'un langage évolué : variables et structures de contrôle. Pour compléter le corpus, nous ajoutons une représentation de la formulation du problème faite par l'utilisateur ainsi qu'une description du plan de traitement élaboré par l'expert de traitement d'images. En effet, les travaux des années 90 ont mis en évidence l'intérêt d'une modélisation des applications sous la forme de plans de traitement pour modéliser l'expertise de conception (Thonnat, 1999).

L'atelier fournit donc les outils nécessaires pour constituer ces trois types de corpus au cours du développement d'applications réelles.

2.3 Vue structurelle de l'atelier

L'architecture de l'atelier est donnée figure 1. L'atelier présente macroscopiquement deux volets : l'un automatique dédié au système à construire et l'autre manuel dédié aux experts. Le développement d'une application formulée par un utilisateur devrait pouvoir se faire indifféremment du côté expert ou du côté système à base de connaissances. Les deux volets partagent pour cela les mêmes représentations dont les ontologies sont notamment les garantes.

L'atelier se compose de l'interface de formulation d'application dédiée aux utilisateurs, d'un atelier de génie logiciel propre au traitement d'images, d'une interface de programmation visuelle, d'une interface de programmation visuelle qui permet de construire des programmes sous la forme de graphes d'opérateurs et

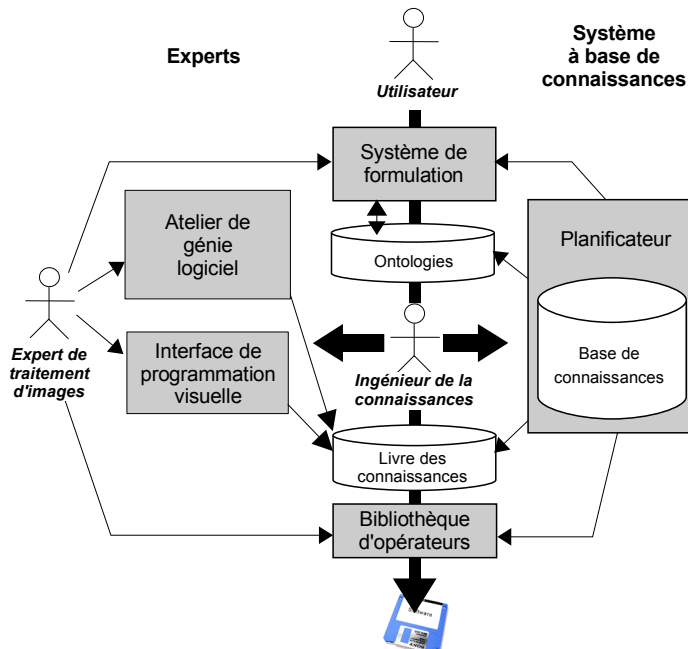


FIG. 1 – L'architecture globale de l'atelier.

d'une bibliothèque d'opérateurs exécutables.

Les corpus sont enregistrés sous la forme d'ontologies, de graphes d'opérateurs et d'un livre des connaissances. La formulation d'application faite par l'utilisateur est capitalisée sous la forme d'une ontologie d'application, le programme sous la forme d'un graphe d'opérateurs et le plan de traitement avec sa logique de conception dans un livre des connaissances (Ermine, 2000).

2.4 Vue fonctionnelle de l'atelier

A l'intérieur de l'atelier, le développement d'application est vu comme un processus itératif et incrémental. L'utilisateur et l'expert de traitement d'images collaborent pour développer une application par l'intermédiaire de l'interface de formulation de problème. L'utilisateur apporte sa connaissance du domaine et ses capacités d'évaluation des résultats produits si ceux-ci sont sous la forme d'images. L'expert apporte sa science du développement d'application. A partir d'une formulation partielle, l'expert ou le système à base de connaissances ou les deux développent une solution qui permet de créer des images de sorties sur lesquels l'utilisateur peut compléter, affiner ou remettre en cause sa formulation. L'important, c'est d'avoir un résultat très rapidement pour ne pas rompre le cycle de formulation. La programmation visuelle est ici un outil précieux : des prototypes exécutables sont créés immédiatement sans les contraintes de la programmation. Enfin l'ingénieur de la connaissance interagit avec l'expert du traitement d'images pour expliciter la logique de conception afin de la capitaliser dans le livre des connaissances.

Le bénéfice dans notre atelier, c'est qu'il n'augmente pas la charge cognitive de l'expert de traitement d'images. Au contraire, les outils permettent d'accroître la production par incitation à la réutilisation et de se recentrer sur la couche métier en se débarrassant des problèmes de programmation et de formalisation.

3 La formulation d'application

De cet atelier nous avons choisi de présenter plus spécifiquement la formulation d'application. La formulation consiste à décrire le problème de traitement d'images à résoudre.

Le rôle du système de formulation, c'est de faire franchir à l'utilisateur les 2 fossés sensoriel et sémantique (Smeulders, 2000) pour passer de sa perception du monde réel à une description à partir de caractéristiques visuelles (figure 2). Le fossé sensoriel résulte de l'écart existant entre le phénomène observé et sa représentation sous forme d'images. La perte d'information est due à l'acquisition d'image. Le fossé sémantique correspond à l'écart entre la représentation sous forme d'image et sa description sous forme d'un ensemble de caractéristiques qui peuvent être symboliques ou numériques. La perte d'information résulte de la représentation linguistique de la dénotation de l'image. Le système de formulation doit donc d'aider l'utilisateur à produire une « bonne » formulation, c'est à dire une formulation qui minimise la perte d'information pertinente et accroît l'abstraction des informations pour qu'elles décrivent au mieux la scène avec le moins d'information possible.

Parce que la formulation d'application de traitement n'a été que peu étudiée en tant que telle, nous avons choisi de construire notre propre modèle conceptuel de la formulation malgré le coût élevé que cela représente. Le modèle conceptuel de la formulation d'application développé par Arnaud Renouf (Renouf, 2007) distingue la spécification des objectifs de la définition de la classe d'images. Les figures 3 et 4 présentent successivement la représentation du modèle conceptuel de la formulation des objectifs et de la classe d'images.

Un atelier d'ingénierie des connaissances en traitement d'images

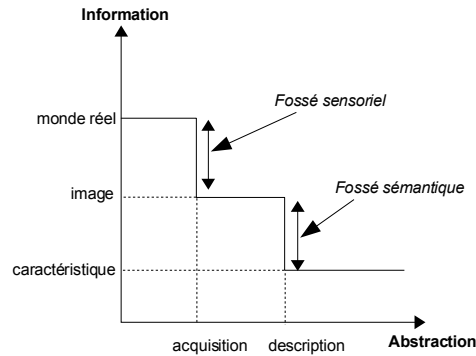


FIG. 2 – Le fossé sensoriel sépare le monde réel de l'image, obtenue par acquisition et le fossé sémantique sépare l'image de sa description linguistique.

Basée sur une analyse sémiotique¹, la classe d'image est décrite sur trois niveaux : le niveau physique porte sur la description des effets de la chaîne de production sur les images, le niveau perceptif décrit le contenu d'image par les caractéristiques des primitives visuelles qui le composent et le niveau sémantique s'intéresse à la description des objets présents dans l'image. Le niveau perceptif est utilisé à défaut ou en complément du niveau sémantique notamment quand les objets sont imprévisibles.

Basé sur une analyse systémique, l'objectif se formule en terme intentionnel sous la forme {tâche, contraintes}. La tâche s'écrit comme une action pouvant porter sur un objet du domaine, une primitive visuelle ou une propriété des images. Les contraintes spécifient les critères à optimiser et les niveaux de détail à atteindre. Chaque contrainte peut être précisée par une erreur acceptable qui permet ainsi de résoudre les compromis en cas de doute sur le respect de la contrainte.

Ce modèle est formalisé dans une ontologie du domaine. Le système de formulation utilise cette ontologie du domaine pour construire avec l'utilisateur une ontologie d'application.

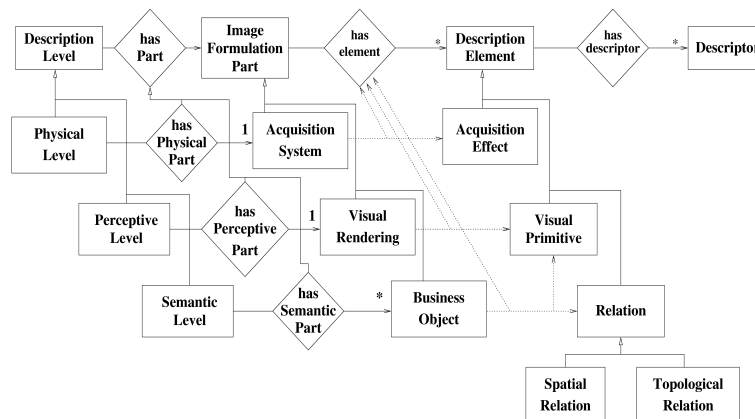


FIG. 3 – Le modèle conceptuel de la définition de la classe d'images décrit les éléments qui composent la description d'une classe d'image et les relations qu'ils entretiennent.

¹ La sémiotique est l'étude des signes et de leur signification.

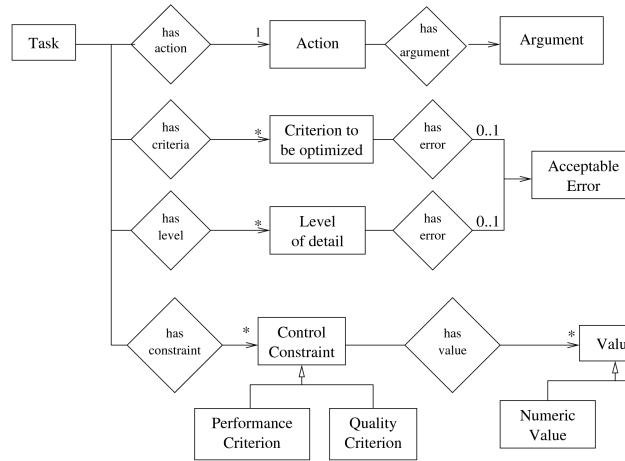


FIG. 4 – *Le modèle conceptuel de la spécification des objectifs de traitement.*

Pour tenter d'objectiver au maximum la formulation de l'utilisateur, le système de formulation met en œuvre un cycle de formulation. Pour illustrer ce cycle nous prenons un exemple d'une application en géographie. Le contexte de l'application est l'étude diachronique de l'utilisation des sols en milieu rural dont une image est présentée figure 5.

Étape 1. Le cycle est initié par la spécification des post-traitements. C'est un point d'entrée assez naturel. Les post-traitements sont décrits par la tâche à opérer et les mesures qui seront pratiquées sur les images de sortie du traitement d'images. Ces post-traitements ne font pas partie à proprement parlé de la formulation de traitement d'images mais ils constituent une source d'information permettant d'aider à la formulation (cf. étape 4).

Exemple : Le post-traitement concerne la comparaison d'images et les mesures pratiquées sont de type radiométrique, calcul de surface et localisation.



FIG. 5 – *Les images proviennent de la base de données Google Map et visualise le paysage bocagier de la région de Bayeux (Calvados).*

Étape 2. L'utilisateur renseigne ensuite les effets de la chaîne de production de l'image. Il est bien question des effets et non de la description des éléments de la chaîne. En effet seule une description phénoménologique des effets de la chaîne est utilisable pour orienter les traitements. De plus, ces informations sont relatives au domaine du traitement d'images et indépendante de l'évolution des capteurs.

Exemple : Les images sont altérées par l'effet de blocs du à la compression JPEG.

Étape 3. La définition du domaine est basée sur la construction de l'arbre des objets du domaine. Ce choix repose sur l'hypothèse que les spécialistes d'un domaine partagent une même taxonomie des objets de leur domaine. L'arbre fige les relations de composition entre objets (relation « partie-de »).

Exemple : la figure 6 donne l'arbre des objets présents dans les images.

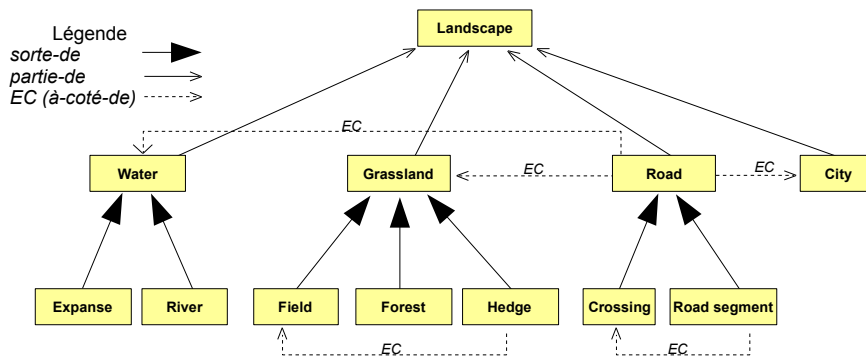


FIG. 6 – L'arbre des objets géographiques.

Étape 4. Il est temps ensuite de spécifier les tâches de traitement à opérer et les contraintes à y associer. Une liste des tâches et de contraintes est fournie par l'ontologie et reprend de façon plus fine les six catégories pointées par notre définition du domaine du traitement d'images. Les post-traitements donnés plus tôt peuvent être utilisés ici pour aider à la spécification des contraintes. Par exemple, si des mesures de radiométrie sont à opérer lors des post-traitements alors une contrainte sur un objet de segmentation devrait être de localiser avec précision la frontière des régions et en cas de doute préférer mettre la frontière à l'intérieur plutôt qu'à l'extérieur pour éviter des erreurs de mesures en prenant des pixels n'appartenant pas à la région.

Exemple : tâche : détection les villes.

Critère à optimiser : maximiser la détection (préférer des fausses alarmes aux oublis)

Niveau de détail : inclure tous les pixels dans la région (préférer plus de pixels que moins)

Étape 5. Sur la base de l'arbre des objets, l'utilisateur doit à présent décrire chacun des objets identifiés par la caractérisation de la primitive visuelle par laquelle il se manifeste (e.g., région, contour, point d'intérêt). Il est fait ici usage du paradigme différentiel qui veut que la description des objets se fasse par différence avec les autres objets. Si deux objets existent dans l'arbre c'est qu'ils présentent au moins une caractéristique différente. L'essentiel de la description d'un objet est fait par les caractéristiques qui le discriminent des autres. La liste des primitives et de leurs caractéristiques est fournie par l'ontologie.

- Computer Vision and Image Understanding*, 67(2):161-185.
- Charlet, J. *L'ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales*. Mémoire d'habilitation à diriger les recherches, Université Pierre et Marie Curie, Paris, 2002.
- Draper, B., A. Hanson, A. E. Riseman (1996), Knowledge-Directed Vision : Control, Learning, and Integration, *Proc. of IEEE*, 84(11):1625-1681.
- Ficet-Cauchard V. , M. Revenu , C. Porquet , R. Clouard (2000), Aide à la conception d'applications de traitement d'images : une approche fondée sur le raisonnement à partir de cas. *Ingénierie des connaissances: évolutions récentes et nouveaux défis*, Eyrolles, 93-108.
- Ermine, J-L., *Les systèmes de connaissances* (2e édition), Hermès, Paris, 2000.
- Gaines B.R., M.L.G. Dhaw, J.B. Woodward (1993). Modelling as framework for knowledge acquisition methodologies and tools. *Int. journal of Intelligent Systems*, 8(2):155-168.
- Garbay, C. (2002). Pour une conception distribuée des systèmes de vision, *Revue L'Objet*, Numéro Spécial « Systèmes distribués et connaissances », 8(4).
- Hanson, A. and E. Riseman (1978). VISIONS: A computer System for Interpreting Scenes. *Computer Vision Systems*, New York: Academic Press, 303-334
- Labiche, J., E. Trupin, Y. Saidali, D. Dionisi, M. Holzem, J.L. Boudon (2007). De l'analyse à l'interprétation d'images de documents. *7ième journées francophones Extraction et Gestion des Connaissances, Actes de l'atelier ECOI*, 13-24.
- Matsuyama, T. et V. Hwang (1990). *SIGMA: A Knowledge-Based Aerial Image Understanding System*. Perseus Publishing.
- Linster M. (1993). Closing the gap between modeling to make sense and modeling to implement systems. *International Journal of Intelligent Systems*, 8(2):209-230.
- Renouf A. (2007). *Modélisation de la formulation d'applications de traitement d'images*, Thèse de Doctorat, Université de Caen.
- Smeulders, A. W. M., M. Worring, S. Santini, A. Gupta, R. Jain(2000), Content-Based Image Retrieval at the End of the Early Years, *IEEE PAMI*, 22:1349-1380.
- Saidali, Y., E. Trupin, M. Holzem, N. Baudoin (2007). Pour une aide à l'interprétation de connaissances traiteur d'images: une approche terminologique. *7ième journées francophones Extraction et Gestion des Connaissances, Actes de l'atelier ECOI*, 25-37.
- Thonnat M., S. Moisan et M. Crubézy, (1999). Experience in integrating image processing programs. *Proceedings of the 1st Int. Conference on Vision Systems*, 200-215.

6 Summary

Our goal long-term research is to develop a knowledge-based system that can automatically generate image processing software from a specification made by a user. This objective has already been widely studied and met with a setback. However, after analyzing the causes of this failure, we propose to take this research from a new perspective: the constructivist approach. For that we need to build an image processing engineering knowledge workbench that is the place of the study and development of such a software. The workbench is a set of software tools designed to encourage the collaboration between the expert image processing and engineering knowledge in order to define an application development theory from the point of view of the system.