Reconnaissance et Extraction de Pièces d'identité

Olivier Augereau, Nicholas Journet, Jean-Philippe Domenger

Reconnaissance et Extraction de Pièces d'identité

Une application industrielle à la détection de cartes d'identité et de passeports

Olivier Augereau, Nicholas Journet, Jean-Philippe Domenger

Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France.
CNRS, LaBRI, UMR 5800, F-33400 Talence, France.
olivier.augereau@labri.fr

RÉSUMÉ. *Cet article aborde le problème de la reconnaissance et de la localisation de pièces d'identité positionnées à l'intérieur de documents numérisés. L'objectif est de détecter si une pièce d'identité est présente, de la localiser précisément et d'extraire la zone d'intérêt qui la contient pour la fournir à un OCR. Dans un premier temps, un modèle de pièce d'identité est défini et un ensemble de points d'intérêts sont extraits de ce modèle. Puis un ensemble de points d'intérêts est extrait de l'image à analyser, cet ensemble est mis en correspondance avec celui du modèle. Cette étape de mise en correspondance permet de calculer la transformation géométrique (translation, rotation, zoom) permettant le recalage entre le modèle et l'image à analyser. La position précise est finalement déterminée à l'aide de cette transformation. La pièce d'identité est alors extraite.*

ABSTRACT. *This article deals with the problem of recognition and location of identity documents within a scanned image. The aim is to detect if an identity document is present in order to locate and extract the region of interest containing it and then to provide the region to an OCR. Initially, a model of an identity document is defined and a set of points of interest are extracted from this model. In a second step, a set of points of interest is extracted from the query image. This set is matched with the model set. This matching is used to calculate the geometric transformation (translation, rotation, zoom) allowing the registration between the model and query image. The position is determined by using the transformation matrix, so the identity can be extracted.*

MOTS-CLÉS : *point d'intérêts, FLANN, SURF, RANSAC, recherche de sous-image.*

KEYWORDS: *interest points, FLANN, SURF, RANSAC, sub-image retrieval*

1re soumission à *CIFED 2012*, le 8 décembre 2011

1. Introduction

Cet article se place dans un contexte industriel. Les entreprises de dématérialisation souhaitent déterminer si des pièces d'identité telles que des cartes d'identité ou des passeports sont présentes sur une page numérisée. Ce que nous appelons "image" dans la suite de l'article est une page numérisée. Une image peut contenir un ou plusieurs documents de différentes natures. Quelques exemples d'images à analyser sont présentés sur la figure 1. Si une pièce d'identité est présente, elle doit être extraite afin d'exploiter les informations qu'elle contient tel que le nom, le prénom, etc.

Les pièces d'identité ont différentes particularités. Tout d'abord, leur structure est rigide, les informations ont une position définie. Les pièces d'identité contiennent une photo d'identité, des logos, des textures et du texte. La dernière particularité est que la plupart des pièces d'identité sont protégées contre les contrefaçons, certaines informations (comme des hologrammes) disparaissent ou sont déformées lors de la numérisation. Une autre particularité rendant complexe la localisation et l'extraction des documents d'identité, est qu'ils peuvent être placés d'une manière quelconque sur l'image (différents orientations et positions). Certains exemples sont visibles sur la figure 1. Il peut également arriver que l'image du document ne soit pas à l'échelle standard et que certaines parties soient coupées ou recouvertes par d'autres informations. De plus il peut y avoir des documents de différentes types sur une même image. Notre problématique se rapproche donc de la recherche de sous-image (*sub-image retrieval*). Les contraintes industrielles nous pousseront à choisir une méthode privilégiant la précision au rappel : en effet il est préférable que les décisions prises par l'algorithme de localisation soient correctes le plus souvent possible. Mieux vaut ne pas détecter certains documents présents dans l'image plutôt que faire de mauvaises détections.

Notre problématique consiste à reconnaitre et à extraire des portions d'images de documents. Cette problématique a été abordée dans le cadre de la recherche dans des bases d'images naturelles. Des propositions ont été faites pour permettre d'aligner des images les unes par rapport aux autres (Yang *et al.,* 2007a), pour rechercher des logos (Rusinol *et al.,* 2009), (Psyllos *et al.,* 2010) ou encore pour créer des panoramas (Brown *et al.,* 2007). Toutes ces techniques sont basées sur la détection de points d'intérêts. Cette méthodologie consistant à détecter des points d'intérêts a également été testée sur des images de documents. On peut notamment citer (Smith *et al.,* 2011) qui s'appuient sur des points d'intérêt pour améliorer la reconnaissance optique de caractère en milieu fortement bruité. Les auteurs de (Takeda *et al.,* 2011) présentent une méthode basée sur des points d'intérêt permettant de retrouver un document en temps réel dans une base de données d'un million de documents. L'inconvénient est que cette méthode est entièrement basée sur le texte. Les points extraits sont les centres des mots. Il faut donc que le texte soit segmenté et les lettres regroupées efficacement. Cette méthode donne de moins bons résultats lorsque des illustrations sont présentes sur les documents.

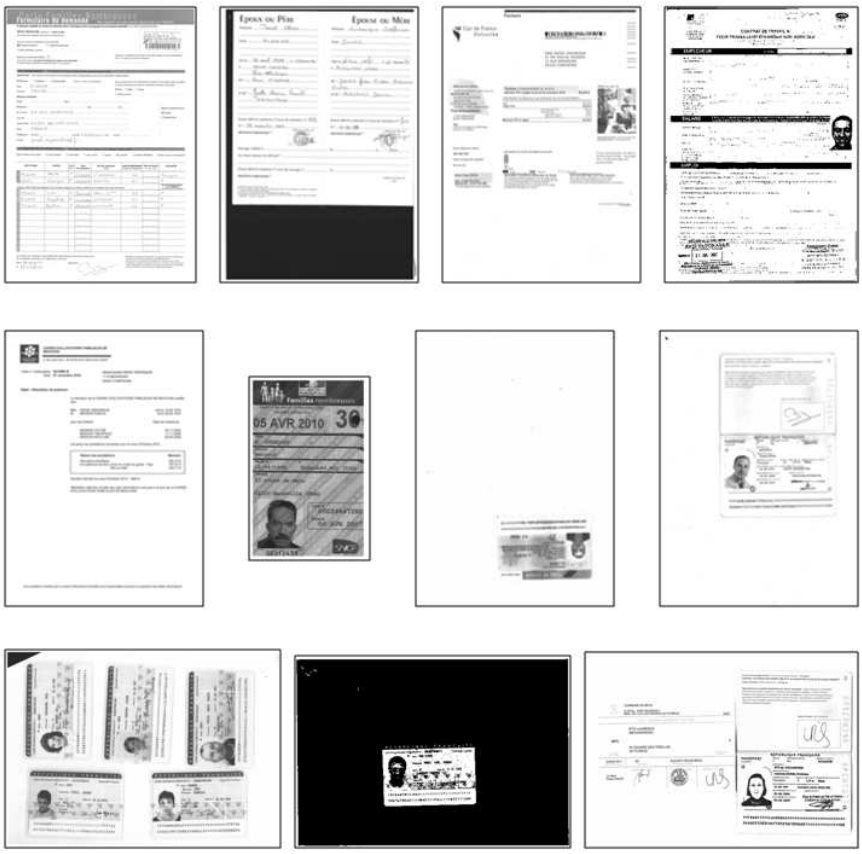

Figure 1. *Exemples de documents présents dans la base (de gauche à droite et de haut en bas) : un formulaire, un acte de mariage, une facture, un contrat, une attestation, une carte d'abonnement, un titre de séjour, un passeport français, cinq cartes d'identité françaises, une carte d'identité française et un passeport français accompagné d'un autre document.*

L'algorithme que nous proposons pour la reconnaissance et l'extraction de pièces d'identité est décrit dans la figure 2. Dans un premier temps, une image modèle correspondant au type de pièces d'identité à reconnaitre est créée. Une image modèle correspond à un exemple de document recherché pour lequel les parties variables ont été supprimées. Les points d'intérêt des différents modèles et des images requêtes sont extraits et décrits à l'aide de la méthode SURF (Bay *et al.,* 2008). Ensuite les points d'intérêt du modèle sont mis en correspondance avec les points d'intérêt de l'image requête. Pour cela, l'algorithme de recherche rapide et approximative de plus proches voisins FLANN (Muja *et al.,* 2009) est utilisé. Enfin la transformation (modèle à 4

paramètres) est estimée avec RANSAC (Fischler *et al.,* 1981). Cette opération permet de localiser très précisément le modèle dans l'image requête.

Dans la prochaine section nous détaillerons chacune des parties de notre algorithme. Ensuite, nous exposerons l'ensemble des tests effectués et commenterons les résultats obtenus sur différentes bases d'images réelles. Enfin nous discuterons des conclusions et perspectives.

2. Mise en correspondance de pièces d'identité

La mise en correspondance d'images se décompose en trois parties principales : 1) la détection de points d'intérêt, 2) la caractérisation de ces points et enfin, 3) la mise en correspondance des points. Dans notre cas, une quatrième étape d'estimation de transformation géométrique est ajoutée car le document doit être localisé précisément afin d'être extrait.

Dans un premier temps, le modèle qui servira à la comparaison est créé.

2.1. *Création du modèle*

Le modèle est créé à partir d'une image existante. Pour cela il faut se placer dans les meilleures conditions possibles et choisir une image de bonne qualité afin d'être certain que des points d'intérêt seront détectés sur les zones intéressantes de l'image. Puisque nous utilisons une technique de recherche exacte, les parties qui varient d'une image à l'autre peuvent être enlevées. En effet, garder les zones variant d'un document à l'autre ralentiraient inutilement le traitement en ajoutant plus de points d'intérêt à trouver. De plus, cela permet de minimiser le nombre potentiel de mauvaises mises en correspondance. Concrètement, pour une carte d'identité nous enlevons toutes les informations variables relatives aux personnes (nom, prénom, adresse,...). La figure 3 représente un exemple de modèle de carte d'identité française.

2.2. *Détection et description de points d'intérêt*

Cette étape consiste à détecter des points d'intérêt dans les images à comparer. Les points sont caractérisés par la configuration de leur voisinage. Le descripteur associé au point doit être le plus robuste possible afin d'être reconnu même si la position et l'orientation du document diffèrent entre les deux images à comparer.

La plupart des détecteurs existant sont invariants en translation, comme par exemple le détecteur de Harris (Harris *et al.,* 1988). Les évolutions telles que Harris-Laplace ou les méthodes basées sur les DOG (Difference-of-Gaussian) sont quant à elles invariantes en rotation et en changement d'échelle. Les techniques telles que MSER (Matas *et al.,* 2004) et LLD (Cao, 2008) ont été mises au point dans l'objectif d'être invariantes aux transformations affines. Cependant, SIFT reste la référence en

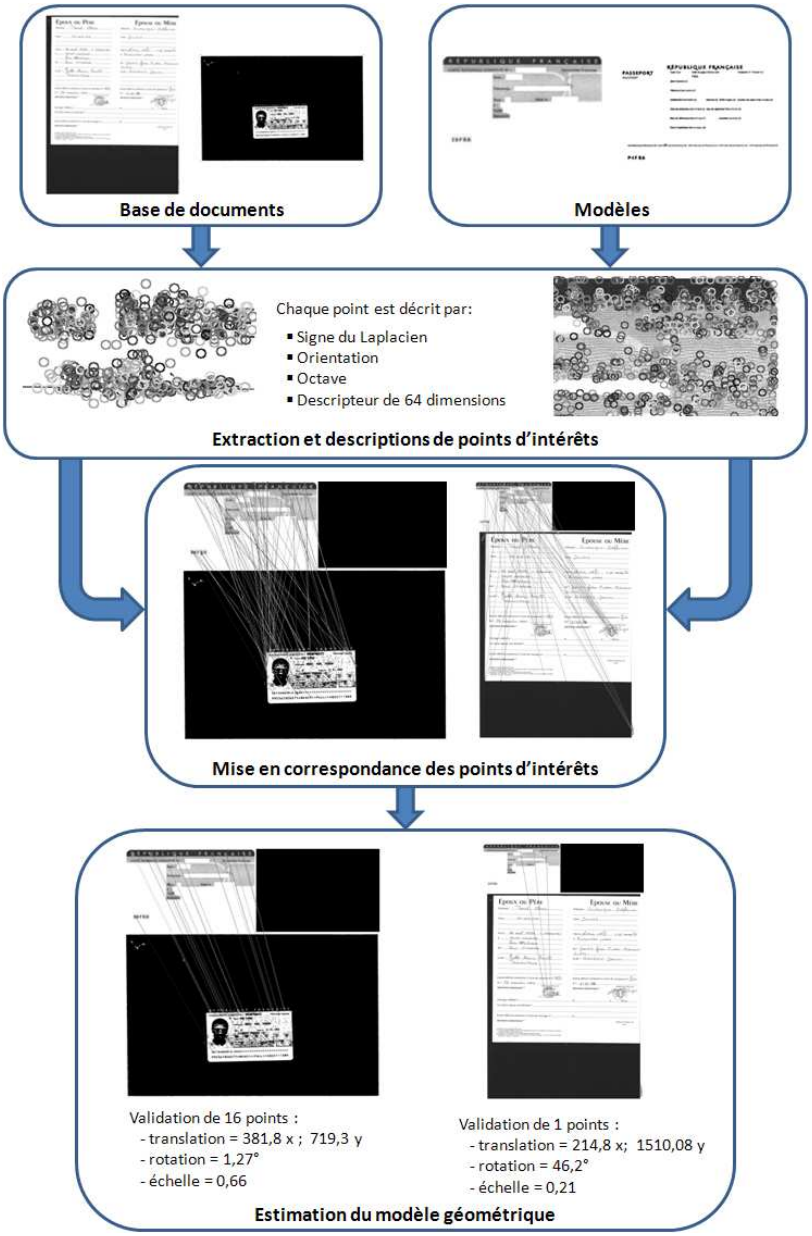

Figure 2. *Les points d'intérêt du modèle et de l'image requête sont extraits et caractérisés avec SURF puis mis en correspondance avec FLANN. La transformation géométrique est ensuite estimée à l'aide de RANSAC.*

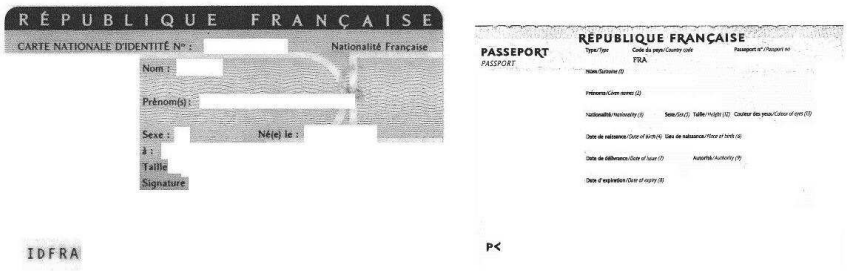

Figure 3. *Création d'une image modèle de carte d'identité. Les parties variables de la carte d'identité sont mises en blanc afin de ne pas trouver de points d'intérêt dans ces zones.*

matière de détection de points d'intérêt. Il combine les DoG qui sont invariants en translation, rotation et mise à l'échelle avec un descripteur basé sur les distributions d'orientations de gradient qui de plus est robuste aux changements d'illumination et de points de vues. Depuis, quelques variantes et extensions de SIFT telles que PCA-SIFT (Ke *et al.,* 2004), ASIFT (Morel *et al.,* 2009) et SURF ont été mises au point.

Le descripteur SURF est principalement reconnu pour sa rapidité de calcul. L'étude comparative (Juan *et al.,* 2010) démontre la supériorité du descripteur SURF par rapport à SIFT et PCA-SIFT d'un point de vue de ces performances en temps d'exécution et de sa robustesse aux changements d'illumination. L'algorithme SURF est composé de deux étapes principales. La première consiste à détecter des points d'intérêt sur l'image et la seconde consiste à décrire ces points d'intérêt à l'aide d'un vecteur de 64 caractéristiques.

2.2.1. *Détection des points*

Afin de gagner du temps de calcul, l'image à analyser est transformée en image intégrale (Viola *et al.,* 2001). Les images intégrales permettent de faire beaucoup plus rapidement les calculs de convolution et d'aires rectangulaires. Soit i, notre image de départ, $i(x, y)$ représente la valeur d'un pixel de l'image aux coordonnées x et y. L'image intégrale, notée $ii(x, y)$, est une image de même taille que l'image d'origine, calculée à partir de celle-ci. Chaque pixel de l'image intégrale contient la somme des pixels situés au dessus et à gauche de ce pixel dans l'image initiale. La valeur d'un pixel de l'image intégrale ii est définie à partir de l'image i par l'équation suivante :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Une fois l'image intégrale calculée, l'aire $Aire$ d'un rectangle $ABCD$ de l'image d'origine peut être évaluée en accédant seulement à la valeur des quatre sommets alors qu'il faudrait accéder à toutes les valeurs des pixels du rectangle sans image intégrale :

$$Aire = \sum_{\substack{x_C \leq x' \leq x_D \\ y_C \leq y' \leq y_D}} i(x', y') = ii(C) + ii(D) - ii(B) - ii(A)$$

Les zones de fort changement d'intensité des pixels sont recherchées dans l'image. La matrice Hessienne, basée sur le calcul des dérivées partielles d'ordre deux, est utilisée pour cela. Pour une fonction à deux variables $f(x, y)$, la matrice hessienne est définie comme suit :

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Si le déterminant de la matrice Hessienne est positif, alors les valeurs propres de la matrice sont toutes les deux positives ou toutes les deux négatives, ce qui signifie qu'un extremum est présent. Les points d'intérêt seront donc localisés là où le déterminant de la matrice Hessienne est maximal. Concrètement, les dérivées partielles du signal sont calculées par un produit de convolution avec des gaussiennes. Afin de gagner en rapidité, ces gaussiennes sont approximées par une fonction à palier appelée *box filter*.

Une représentation à des niveaux d'échelles plus bas est obtenue en augmentant la taille des filtres gaussiens. Au final, les points d'intérêt dont le déterminant de la matrice hessienne est positif et qui sont maximum locaux dans un voisinage 3 x 3 x 3 (abscisse x ordonnée x échelle) sont conservés.

2.2.2. *Description des points*

Une fois les points d'intérêt extraits, la seconde étape de SURF consiste à calculer le descripteur correspondant. Le descripteur SURF décrit l'intensité des pixels dans un voisinage autour de chaque point d'intérêt. La réponse en x et en y des ondelettes de Haar est calculée dans un voisinage de $6s$ où s est l'échelle à laquelle le point d'intérêt a été trouvé. A partir de ces valeurs, l'orientation dominante de chaque point d'intérêt est calculée en faisant glisser une fenêtre d'orientation.

Pour calculer le descripteur, un carré de taille $20s$ orienté selon l'orientation dominante est extrait. Cette zone est subdivisée en 4×4 carrés. Pour chacune de ces sous-régions, les ondelettes de Haar sont calculées sur 5×5 points. Soit d_x et d_y la réponse à l'ondelette de Haar, 4 valeurs sont calculées pour chacune des sous-régions : $\sum d_x$, $\sum d_y$, $\sum |d_x|$ et $\sum |d_y|$. Au final, chacun des points extraits à l'étape précédente est décrit par un vecteur composé de $4x4x4$ valeurs soit 64 dimensions.

2.3. *Mise en correspondance des points*

Dans cette étape, les points d'intérêt d'une image sont mis en correspondance avec les points d'intérêt d'une autre image afin d'estimer le degré de similitude entre ces deux images. Chaque point d'intérêt de l'image modèle est associé aux deux points d'intérêt de l'image requête les plus proches. Le plus proche voisin serait suffisant, mais le second plus proche sera utilisé pour l'étape de filtrage qui suit. Pour trouver les 2-PPV (deux plus proches voisins) la distance euclidienne entre les descripteurs à 64 dimensions est utilisée.

La recherche de PPV peut s'avérer longue si elle est faite de manière exhaustive. Les arbres KD permettent de structurer l'espace de recherche afin d'accélérer la comparaison d'un élément avec les autres. La figure 4 illustre la construction et la recherche de plus proches voisins avec un arbre KD. Les performances de recherche d'un arbre KD se rapprochent de celles d'une recherche linéaire lorsque le nombre de dimensions est grand. Un arbre KD ne serait donc pas performant pour SURF puisqu'il y a 64 dimensions. L'algorithme FLANN décrit dans (Muja *et al.,* 2009) utilise le principe d'arbres KD aléatoires proposé récemment par (Silpa-Anan *et al.,* 2008). Dans ce cas, n arbres KD sont utilisés en parallèle, chacun utilisant uniquement 5 dimensions tirées aléatoirement.

Après la mise en correspondance, chacun des points du modèle est associé aux deux points les plus ressemblants dans l'image requête. On souhaite éliminer le plus de fausses mises en correspondance possibles afin de faciliter l'estimation de la transformation. Le premier filtre consiste à supprimer les mises en correspondance dont les 2-ppv sont trop proches l'un de l'autre. C'est le filtrage par unicité, il permet d'éliminer les mises en correspondance ambiguës. Ensuite les mises en correspondance sont filtrées en fonction de l'échelle et l'orientation. Le rapport de l'échelle et la différence d'orientation des mises en correspondance sont calculés. L'espace des angles est découpé par tranche de 20 degrés et l'espace des échelles par facteurs de $1,5$. Les correspondances dont l'échelle et la rotation ne correspondent pas au vote de l'échelle et l'angle majoritaire, sont éliminées. La figure 5 met en évidence l'intérêt de ces filtrages qui permettent d'éliminer facilement un grand nombre de mauvaises mises en correspondance.

2.4. *Estimation de la transformation géométrique*

Dans notre cas, les documents sont numérisés à plat, il n'y a pas de distorsion ni de rotation autre que celle dans le plan. Le modèle recherché comporte quatre inconnues : l'angle θ de rotation dans le plan, la translation T_x selon l'axe x, la translation T_y selon l'axe y et la mise à l'échelle α (uniforme en x et y). La matrice de transformation recherchée M_t est de la forme suivante :

$$M_t . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha.cos(\theta) & -sin(\theta) & T_x \\ sin(\theta) & \alpha.cos(\theta) & T_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

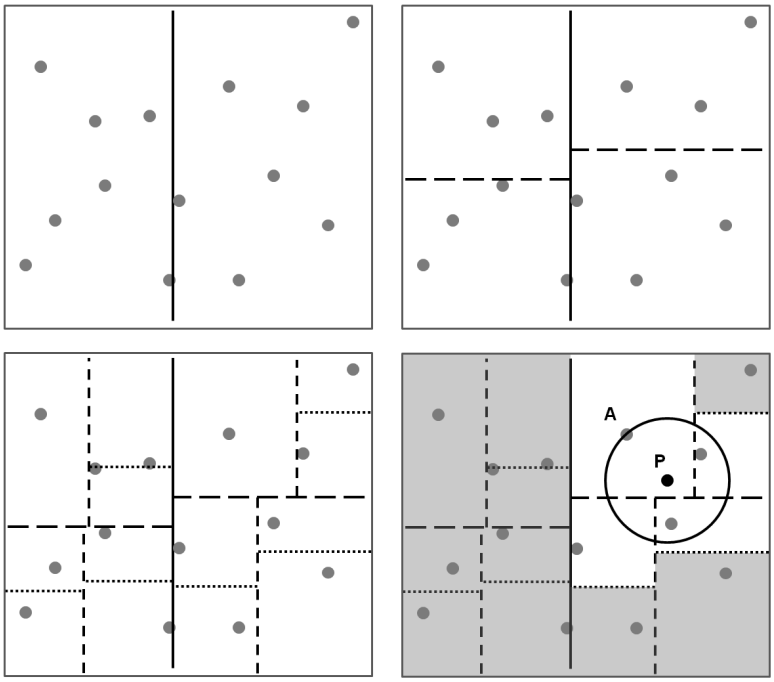

Figure 4. *L'espace est récursivement découpé en hyperplans, à hauteur de la médiane selon une dimension (x) puis selon la dimension suivante (y), etc. L'arbre KD est l'arbre binaire correspondant au découpage. La première partie de la recherche consiste à déterminer dans quelle case de l'arbre est le point P dont le PPV est cherché. Le point A présent dans cette case est marqué comme étant l'actuel PPV. Toutes les cases qui sont plus loin que l'hypersphère de centre P et de rayon PA sont éliminées. Il ne reste plus qu'à trouver si un point dans les cases restantes est plus près de P que A.*

La difficulté de cette partie est de trouver une transformation géométrique parmi les différentes mises en correspondance alors qu'il reste des mises en correspondance erronées et que le document recherché peut ne pas être présent sur l'image.

Pour ces deux raisons la méthode des moindres carrés ne peut pas être utilisée car elle serait perturbée par les mises en correspondance erronées (valeurs aberrantes) et trouverait une transformation même quand le modèle n'est pas présent. L'estimation de la transformation géométrique M_t doit donc être faite à l'aide d'un algorithme capable d'estimer un modèle sans prendre en compte les valeurs aberrantes (aussi appelées outliers).

Pour cela il existe des méthodes robustes basées sur les statistiques telles que les LmedS (Least Median of Squares) (Rousseeuw, 1984), les M-estimateurs (Huber *et*

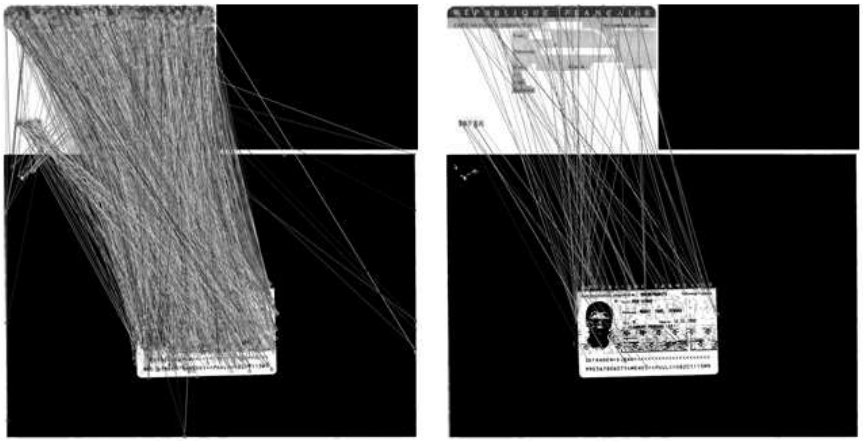

Figure 5. *Filtrage des points d'intérêts. À gauche :* 2438 *mises en correspondance sont faites (une pour chaque point d'intérêt). À droite : il reste 78 mises en correspondance après filtrage par unicité, angle et échelle.*

al., 1981) ou encore la transformée de Hough.Une étude comparative de ces algorithmes peut être trouvée dans (Malis *et al.,* 2005). Selon (Sur *et al.,* 2011), RANSAC ((Chum *et al.,* 2003), (Chum *et al.,* 2005)) est la méthode la plus utilisée.

En résumé, l'algorithme RANSAC s'articule en deux étapes principales. 1) Le sous ensemble le plus petit possible est sélectionné aléatoirement afin de pouvoir estimer le modèle géométrique. 2) Les éléments validant le modèle sont cherchés. Ces éléments sont appelés "*inliers*". S'il n'y a pas suffisamment d'inliers, l'algorithme retourne en 1), sinon le modèle est validé. Si aucun modèle n'est validé après un certain nombre d'essais l'algorithme est arrêté. Une application simple de RANSAC à la détection d'une droite est illustrée sur la figure 6.

Le fonctionnement de RANSAC détaillé et appliqué à notre cas est le suivant :

1) Deux mises en correspondance sont tirées au hasard. Soit A et B les points respectivement de la première et seconde mise en correspondance dans l'image modèle et A' et B' les correspondances dans l'image requête. Si les distances $||AB||$ ou $||A'B||'$ sont trop petites, deux nouvelles mises en correspondance sont tirées. Cette condition est rajoutée car les points peuvent être confondus, ce qui donnerait une fausse estimation du modèle. De plus, si les points sont trop proches, le modèle géométrique risque d'être mal estimé. La distance minimale entre deux points à été fixée à 5.

2) La matrice M_t correspondante est calculée. θ est égal à l'angle formé par $\overrightarrow{AB}$ et $\overrightarrow{A'B'}$, α est égal au rapport de $||AB||$ sur $||A'B'||$. T_x et T_x sont déterminées en projetant $\overrightarrow{AA'}$ sur $\overrightarrow{x}$ et sur $\overrightarrow{y}$.

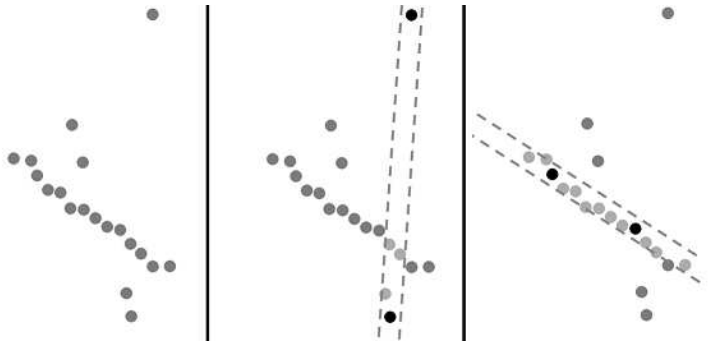

Figure 6. *Application de RANSAC à la détection d'une droite. Les points noirs sont les points sélectionnés au hasard pour estimer le modèle de la ligne droite. Les lignes en pointillés représentent la marge d'erreur tolérée par le modèle. Les points en gris clair sont les inliers validés par le modèle.*

3) Pour chacune des autres mises en correspondance, le nombre d'éléments validant ce modèle est calculé. Un point est valide si la distance entre la transformation de ce point par la matrice M_t et la mise en correspondance de ce point est inférieur à un seuil donné. Dans notre cas le seuil est fixé à une distance de 10 pixels. Pour éviter que trop d'inliers soient trouvés dans une seule petite zone de l'image, pas plus d'une mise en correspondance dans un rayon de 5 pixels n'est validée.

4) Si un certain nombre de points sont valides alors le modèle de transformation testé est validé. Sinon : si le nombre de tentatives est inférieur à MAX_ITER, le nombre de tentatives est incrémenté et l'algorithme retourne en 1) ; sinon le modèle n'est pas considéré comme présent. Dans notre cas, $MAX\_ITER = 1000$.

Si le modèle est validé, la matrice de transformation est alors connue. La sous-image correspondante peut être extraite. La figure 7 illustre le résultat de l'application de RANSAC.

3. Tests sur bases réelles

3.1. *Recherche de pièces d'identité en niveaux de gris et noir et blanc.*

Les images de la première base BD_NB ont été numérisées et binarisées à la volée afin d'être stockées en fichier TIF. Dans cette base, la plupart des images sont au format A4 et toutes ont une résolution de 200 dpi. Chaque image ne contient qu'un seul document. Cette base contient 483 cartes d'identité françaises, 89 passeports français, et 508 images de différents types autres que les deux types de pièces d'identité recherchés. Soit un total de 1080 images.

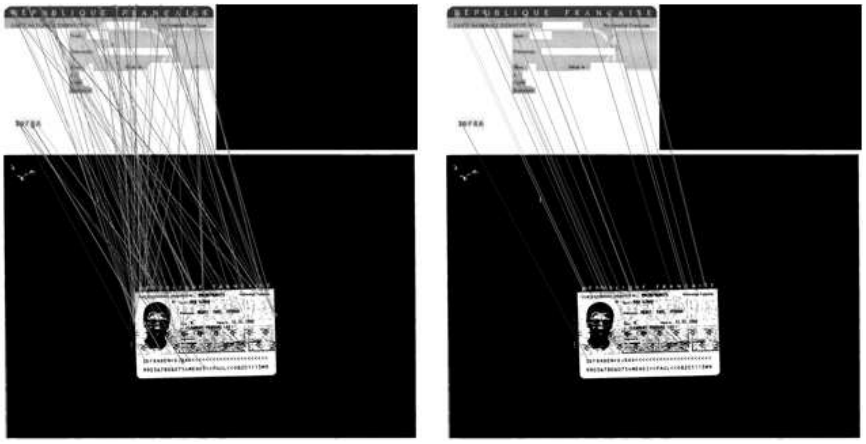

Figure 7. *Estimation du modèle géométrique avec RANSAC. À gauche : il reste 78 mises en correspondance après filtrage par unicité, angle et échelle. À droite : le meilleur modèle trouvé par RANSAC valide 16 mises en correspondance.*

Pour la recherche d'un type de document T_i particulier, les métriques rappel et précision sont calculées. Le rappel représente le rapport du nombre de documents correctement attribués à T_i sur le nombre de documents appartenant à T_i. La précision est égale au nombre de documents correctement attribués à T_i sur le nombre total de documents attribués à T_i.

Puisque la base de données est en noir et blanc, les modèles créés pour la recherche de cartes d'identité françaises et de passeports français ont été construits à partir d'une image en noir et blanc. Les tests ont été effectués sur un ordinateur Intel Core 2 Duo, 2 GHz. Le tableau 1 représente les résultats de la recherche de cartes d'identité et de passeports dans la base BD_NB.

Tableau 1. *Performances de détection de pièces d'identité, sur la base d'image de documents noir et blanc (N&B) BD_NB avec des images modèles en noir et blanc. Aucune mauvaise détection n'est faite et au moins 96% des cartes d'identité ou des passeports sont retrouvés. Avec des images modèles en niveaux de gris (NdG), aucune mauvaise détection n'est faite et au moins 76% des documents sont retrouvés.*

Modèles	Rappel	Précision	Temps moyen par image
Carte Id (N&B)	0,96	1	7s
Passeport (N&B)	0,97	1	9s
Carte Id (NdG)	0,76	1	8s
Passeport (NdG)	0,85	1	9s

Les images de la seconde base ont été numérisées en niveaux de gris, à une résolution de 300 dpi. Les images sont différentes de celles de la première base. Chaque image est composée également d'un seul document. BD_NdG contient 91 cartes d'identité françaises, 17 passeports français, et 449 images de documents de types quelconques. Soit un total de 557 images. Le tableau 2 montre qu'il est important d'avoir un modèle de référence qui soit de même nature que les images requêtes.

Tableau 2. *Performances de détection de pièces d'identité, sur la base d'images de documents en niveaux de gris (NdG) BD_NdG avec des images modèles en niveaux de gris. Aucune mauvaise détection n'est faite et au moins 94% des cartes d'identité ou des passeports sont retrouvés. Avec un modèle en noir et blanc (N&B) le nombre de documents détectés baisse.*

Modèles	Rappel	Précision	Temps moyen par image
Carte Id (NdG)	0,97	1	9s
Passeport (NdG)	0,94	1	14s
Carte Id (N&B)	0,97	1	9s
Passeport (N&B)	0,59	1	14s

La précision de cette technique est excellente car les similarités locales extraites avec robustesse sont validées globalement par un modèle géométrique. Par contre le rappel n'est pas optimum. Les documents non identifiés présentent généralement les défauts suivants : un bruit très marqué et des problèmes de binarisation. Ces problèmes sont dus à la chaine de numérisation : les pièces d'identité sont numérisées puis imprimées par l'utilisateur. Cette impression est ensuite numérisée une nouvelle fois par l'entreprise de dématérialisation. A chaque étape, du bruit est susceptible de s'ajouter. La figure 8 illustre ces deux problèmes.

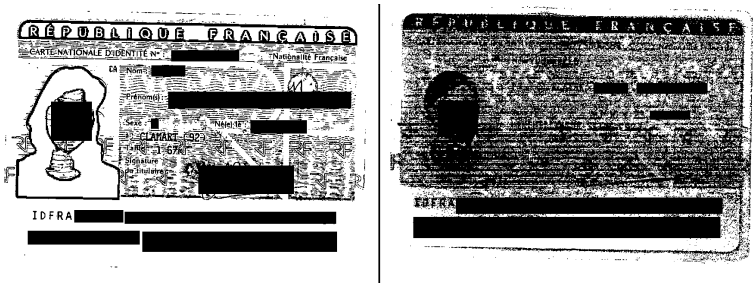

Figure 8. *L'image de gauche présente un contraste inversé notamment dans le bandeau "république française" et l'image de droite un bruit très marqué. Ces deux images ne sont pas détectées.*

3.2. *Problématique de l'auto-similarité.*

Lorsque les images contiennent plus d'une occurrence d'un même élément, la mise en correspondance est fragilisée. Par exemple sur la carte d'identité française est écrit "RÉPUBLIQUE FRANÇAISE". Les lettres "R", "U", "I", "E", "A" sont présentes deux fois et peuvent être confondues. Mais si suffisamment d'autres points d'intérêt sont présents, le document sera tout de même détecté.

Le fait que plusieurs documents de même type soient présents sur une même image rejoint la problématique d'auto-similarité. La figure 9 illustre la méthodologie pour détecter les différents documents. Lorsqu'un modèle est validé, la zone correspondante est supprimée et l'image est traitée de nouveau.

Nous avons testé ce principe sur une base où chaque image contient plusieurs documents de même type. La nature des images de la base BD_Multi est la même que celle de la base BD_NdG. La base est composée d'un total de 32 images contenant 89 pièces d'identité. Le tableau 3 montre que la multi-détection fonctionne bien dans notre cas.

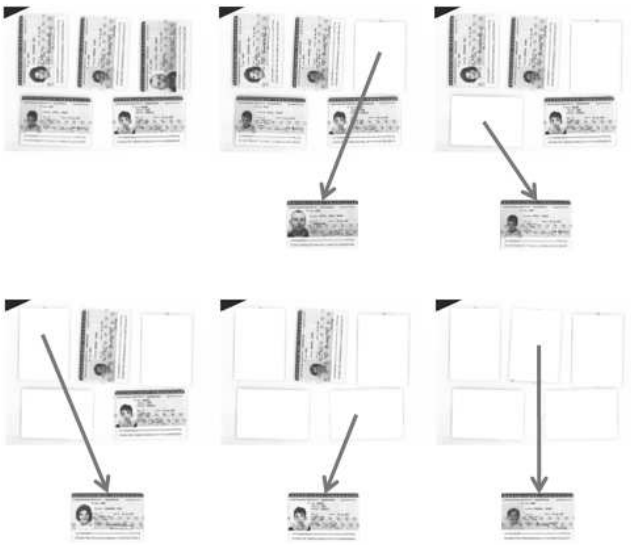

Figure 9. *Multi-détection. On cherche un modèle dans chaque image. Si un modèle est trouvé, il est supprimé de l'image puis on cherche à nouveau si un modèle est présent sur l'image. On réitère jusqu'à qu'on ne trouve plus de correspondance.*

Tableau 3. *Les documents multiples dans les images sont trouvés par l'algorithme. Il peut tout de même manquer des documents pour les mêmes raisons que précédemment : le document manquant est un document très bruité.*

Modèles	Nb d'im	Nb de doc	Nb de doc trouvés	Temps moyen par doc
Carte Id (NdG)	29	83	82	11s
Passeport (NdG)	3	6	6	13s

4. Conclusion

L'identification de sous-parties d'images de documents peut donc se faire très précisément à l'aide de la méthode exposée dans cet article. Le principal intérêt de cette méthode est de garantir une précision de 100%. De plus, la méthode peut être étendue très simplement à la recherche d'autres types de sous-images. Dans le cas d'application à des images contenant du texte, il faudra cependant faire attention au problème d'auto-similarité pouvant être causé par des lettres.

Il y a trois principales perspectives. La première est d'améliorer la rapidité d'exécution. Cela pourrait être fait en utilisant des descripteurs plus légers comme ceux utilisés dans (Takeda *et al.,* 2011). La deuxième perspective vise à intégrer dans notre méthodologie l'usage des sacs de mots visuels (Yang *et al.,* 2007b) afin d'associer un indice de confiance à une décision. La dernière perspective consistera à analyser les techniques telles que (Noury *et al.,* 2010) permettant de traiter des problèmes d'auto-similarité plus complexes.

Remerciement

Nous remercions l'entreprise de dématérialisation Gestform [1], pour le temps passé à nous expliquer leur problématique et à nous fournir des images de documents. Nous remercions également Philippe Larroye, élève à l'ENSEIRB-MATMECA qui a participé dans le cadre d'un stage à la mise au point de l'algorithme.

5. Bibliographie

Bay H., Ess A., Tuytelaars T., Van Gool L., « Speeded-up robust features (SURF) », *Computer Vision and Image Understanding*, vol. 110, n° 3, p. 346-359, 2008.

Brown M., Lowe D., « Automatic panoramic image stitching using invariant features », *International Journal of Computer Vision*, vol. 74, n° 1, p. 59-73, 2007.

Cao F., *A theory of shape identification*, vol. 1948, Springer Verlag, 2008.

1. www.gestform.com

Chum O., Matas J., « Matching with PROSAC-progressive sample consensus », *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, Ieee, p. 220-226, 2005.

Chum O., Matas J., Kittler J., « Locally optimized RANSAC », *Pattern Recognition*p. 236-243, 2003.

Fischler M., Bolles R., « Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography », *Communications of the ACM*, vol. 24, n° 6, p. 381-395, 1981.

Harris C., Stephens M., « A combined corner and edge detector », *Alvey vision conference*, vol. 15, Manchester, UK, p. 50, 1988.

Huber P., Ronchetti E., MyiLibrary, *Robust statistics*, vol. 1, Wiley Online Library, 1981.

Juan L., Gwun O., « A comparison of sift, pca-sift and surf », *International Journal of Image Processing (IJIP)*, 2010.

Ke Y., Sukthankar R., « PCA-SIFT : A more distinctive representation for local image descriptors », 2004.

Malis E., Marchand E., « Méthodes robustes d'estimation pour la vision robotique », *Journées nationales de la recherche en robotique, JNRR'05*, Guidel, France, France, 2005.

Matas J., Chum O., Urban M., Pajdla T., « Robust wide-baseline stereo from maximally stable extremal regions », *Image and Vision Computing*, vol. 22, n° 10, p. 761-767, 2004.

Morel J., Yu G., « ASIFT : A new framework for fully affine invariant image comparison », *SIAM Journal on Imaging Sciences*, vol. 2, n° 2, p. 438-469, 2009.

Muja M., Lowe D., « Fast approximate nearest neighbors with automatic algorithm configuration », *International Conference on Computer Vision Theory and Applications (VISSAPP 09), pages 331*, vol. 340, Citeseer, 2009.

Noury N., Sur F., Berger M., « Modèle a contrario pour la mise en correspondance robuste sous contraintes épipolaires et photométriques », 2010.

Psyllos A., Anagnostopoulos C., Kayafas E., « Vehicle logo recognition using a SIFT-based enhanced matching scheme », *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, n° 2, p. 322-328, 2010.

Rousseeuw P., « Least median of squares regression », *Journal of the American statistical association*p. 871-880, 1984.

Rusinol M., Lladós J., « Logo spotting by a bag-of-words approach for document categorization », *2009 10th International Conference on Document Analysis and Recognition*, IEEE, p. 111-115, 2009.

Silpa-Anan C., Hartley R., « Optimised KD-trees for fast image descriptor matching », *Computer Vision and Pattern Recognition*, vol. 0, p. 1-8, 2008.

Smith D., Harvey R., « Document Retrieval Using SIFT Image Features », *Journal of Universal Computer Science*, vol. 17, n° 1, p. 3-15, 2011.

Sur F., Noury N., Berger M.-O., Image point correspondences and repeated patterns, Research Report n°7693, INRIA, July, 2011.

Takeda K., Kise K., Iwamura M., « Real-Time Document Image Retrieval for a 10 Million Pages Database with a Memory Efficient and Stability Improved LLAH », *ICDAR*, 2011.

Viola P., Jones M., « Rapid object detection using a boosted cascade of simple features », *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, IEEE, p. I-511, 2001.

Yang G., Stewart C., Sofka M., Tsai C., « Alignment of challenging image pairs : Refinement and region growing starting from a single keypoint correspondence », *IEEE Trans. Pattern Anal. Machine Intell*, vol. 23, n° 11, p. 1973-1989, 2007a.

Yang J., Jiang Y., Hauptmann A., Ngo C., « Evaluating bag-of-visual-words representations in scene classification », *Proceedings of the international workshop on Workshop on multimedia information retrieval*, ACM, p. 197-206, 2007b.