Sur les seuils de reconnaissance des formes

On shape recognition thresholds

par Pablo MUSÉ, Frédéric SUR, Jean-Michel MOREL

Centre de Mathématiques et de Leurs Applications, École Normale Supérieure de Cachan, 61, avenue du Président Wilson, 94235 Cachan Cedex

résumé et mots clés

Un enjeu important de la recherche de formes dans une base d'images est la définition de seuils non supervisés permettant d'éviter une déferlante de fausses détections, ou, au contraire, des rejets de formes qui auraient dû être reconnues. En prenant comme exemple une méthode de reconnaissance de forme proposée par Lisani [15, 16], nous montrons que l'on peut répondre à la question suivante : étant donnée une forme requête et une base d'images, à partir de quelle distance entre la forme requête et une forme détectée est-on sûrs que la forme est reconnue ? Cette assurance est quantifiée par le nombre de fausses alarmes associé à la paire requête – forme candidate. Cette méthode ne considère pour l'instant que des morceaux de forme et permet pourtant déjà d'aboutir à des détections sûres basées sur un seul morceau de forme.

Codage invariant de forme, nombre de fausses alarmes.

abstract and key words

A significant stake of shape query in databases of images is to define unsupervised thresholds making it possible to avoid a flood of false detections, or, on the contrary, rejections of shapes which should have been recognized. By taking as example a method of shape recognition introduced by Lisani [15, 16], we show that one can answer the following question: given a query shape and a database of images, below which distance between the query and a detected shape is one sure that the shape is recognized? This insurance is quantified by the number of false alarms associated with the pair query shape – candidate shape. Although this method only considers for the moment pieces of shape, it already leads to sure detections based on a single piece of shape.

Invariant shape encoding, number of false alarms.

1. la reconnaissance de formes

La reconnaissance de formes consiste à chercher une forme-requête extraite d'une image dans une base de données composée de formes issues d'une ou plusieurs images. Le sujet a donné lieu à de nombreux travaux, qui tous suivent le schéma suivant :

1. extraction des formes dans les images ;
2. codage invariant des formes ;
3. comparaison des formes ;
4. décision : peut-on dire si oui ou non les formes issues de l'image requête sont présentes ?

Chacune de ces étapes est cruciale, et un mauvais choix pour l'une d'elles compromet la chaîne entière. C'est pourquoi la plupart des algorithmes de reconnaissance de formes sont en fait dédiés à une application ou à un type d'images ou de formes particulier.

Détaillons les possibilités pour chacune des phases (voir [2, 28, 29, 17] pour l'étude d'autres approches et de méthodes marginales).

1.1. extraction des formes dans les images

Cette étape est abordée uniquement par les auteurs décrivant un système complet de reconnaissance de formes. Les formes peuvent être des frontières de régions issues d'une segmentation des images [6], ou de régions homogènes du point de vue de certaines caractéristiques (couleur, texture, *etc.* [30]). Les formes peuvent être considérées comme des nuages de points, d'*edges*, d'*edgels* (qui sont des *edges* avec une direction), de points caractéristiques (par exemple des coins [25]). Notons que ces méthodes d'extraction restent assez grossières, et ne peuvent pas être stables en raison des nombreux paramètres qu'elles nécessitent. Nous verrons dans la suite que des formes extraites de cette façon ne peuvent pas être comparées par des méthodes précises.

À notre connaissance, la seule méthode dans la littérature permettant une extraction fine et robuste est le seuillage des images, qui fournit des silhouettes précises d'objets [22, 19] (le filtre de Canny est aussi précis mais il n'est pas robuste car il peut donner des bords ouverts et il est sensible aux paramètres de lissage et de seuil). Bien sûr, ceci nécessite de restreindre le problème de la reconnaissance de *formes* à celui de la reconnaissance d'*objets*, bien contrastés, sur un fond uniforme.

1.2. codage des formes

Selon la nature des formes extraites, leur codage varie. Dans tous les cas, on cherche un codage invariant sous certaines transformations, si possible robuste au bruit.

Lorsque des *edges* ou des *edgels* ont été extraits pendant la première phase, ces mêmes caractéristiques peuvent être directement utilisées pour le codage. Lorsque ce sont des points caractéristiques (coins, extrema de courbure) qui ont été extraits, des invariants intégro-différentiels peuvent être calculés en ces points [25]. Notons que le calcul des invariants différentiels est en pratique impossible car le calcul des dérivées est particulièrement instable, même après lissage des images.

Considérer seulement des ensembles d'*edges* ou d'invariants locaux entraîne la perte de l'information de localisation spatiale relative. Certains auteurs proposent alors d'utiliser des chaînes d'*edges* [32] ou d'*edgels* [21].

Lorsque les formes sont extraites comme une suite de points ordonnés (une courbe), plusieurs méthodes de codage sont possibles.

Parmi elles, de nombreuses méthodes sont globales, et par conséquent sensibles aux occlusions.

Mokhtarian [19] propose d'utiliser les extrema de courbure dans le *scale-space*. La fonction donnant leur localisation en fonction de l'échelle de lissage est utilisée comme code de la forme considérée. Alvarez *et al.* [3] ont une approche similaire : ils définissent des invariants de formes basés sur l'évolution de la surface et du périmètre sous l'action du *scale-space*. Dans les deux cas, le *scale-space* permet de s'affranchir des « accidents » locaux de la forme.

Des descripteurs peuvent aussi être calculés pour chaque forme [17] : coefficients de Fourier [34], moments invariants [10], etc. Leur inconvénient majeur est de mélanger information globale et locale, et donc d'être sensibles aux occlusions et distorsions locales.

Il est également possible de coder directement le contour de la forme dans un repère normalisé par les moments, ce qui permet un codage invariant. Ce codage présente l'inconvénient d'être instable [7].

Les méthodes locales permettent, elles, de reconnaître des *morceaux* de formes.

Certains auteurs proposent par exemple de coder directement la courbe-forme par une suite de « lettres », ce qui permet d'utiliser ensuite les algorithmes de recherche de motifs et les techniques syntactiques [11]. Le problème de la quantification semble néanmoins difficilement surmontable.

Une description structurelle classique de la forme peut être obtenue en considérant son squelette (bibliographie dans [17]). Mais l'instabilité du calcul du squelette conduit à l'insertion ou à la disparition casuelle de branches et rend le procédé délicat à utiliser.

Le codage le plus raisonnable semble être finalement de considérer les coordonnées des morceaux de la courbe dans différents repères définis par la forme elle-même. Wolfson [32, 31] (suivi par Rothwell [22]) a proposé des repères définis par bitangentes, ce qui est très robuste. La méthode de Lisani [15] l'est encore plus en ne retenant pas les points de tangence comme points de références, mais en construisant un repère à partir de trois tangentes selon deux directions perpendiculaires. Les codages proposés sont à la fois locaux, invariants, et assez robustes. De plus, Lisani utilise un préfiltrage affine invariant (AMSS) permettant une réduction considérable du nombre de codes sans altération significative de la forme.

Il est à remarquer que la plupart des codages mentionnés ne sont invariants que par translation, ou translation-rotation. Par exemple, une carte d'*edgels* est naturellement invariante par translation. Mais une normalisation n'est possible qu'en groupant les *edgels* par deux pour les similitudes. Le coût des méthodes invariantes et sans normalisation est de fait prohibitif.

1.3. comparaison des codes de forme

La plupart du temps, la comparaison des codes de forme consiste tout simplement à évaluer des distances : distance de Mahalanobis si on veut tenir compte de la statistique de la base de recherche, distance L^p, distances de Hausdorff ou de Procuste [26] si on dispose de courbes ou de morceaux de courbes, *etc.*

Lorsque les codes peuvent être vus comme des nuages de points, la méthode la plus utilisée est le *Geometric Hashing*, qui permet de comparer les nuages, à un certain groupe de transformations près [14, 33]. Une première étape de vote permet d'identifier des candidats à l'appariement avec la requête, puis les candidats sont classés selon la distance après un recalage fin. La complexité en temps, la taille de la table de hachage nécessaire au vote et les nombreux paramètres nécessaires sont des obstacles à la mise en œuvre de cet algorithme. La transformée de Hough généralisée [4] est en fait une méthode similaire.

1.4. décision

À notre connaissance, personne n'a encore proposé de méthode générique de décision acceptation/rejet. Généralement, les appariements avec la requête sont seulement ordonnés (par exemple selon une distance), et au mieux l'ordre est basé sur un argument probabiliste (voir par exemple [24]). Le problème du seuil permettant de répondre automatiquement « oui », mais aussi « non », à la question « la forme est-elle présente ? » n'est jamais abordé. Certains auteurs se sont pourtant intéressés à la question des fausses détections dues au hasard (« hallucinating a wrong fit » [27]), allant jusqu'à quantifier le taux de fausses détections [21, 12, 13]. Néanmoins, aucun d'entre eux n'aboutit à une *décision* automatique. De plus, ils sont amenés à poser comme hypothèse l'indépendance des détections pour estimer la probabilité qu'elles puissent avoir eu lieu par hasard. Ceci n'est *a priori* pas vrai et nous allons contourner ce problème en utilisant la linéarité de l'espérance du nombre de détections, comme nous le verrons par la suite.

1.5. notre contribution

Aucune méthode de la littérature n'est à la fois locale, robuste à l'occlusion, invariante par similitude, et n'extrait les formes d'une image quelconque, tout en donnant des seuils. Nous avons mentionné que Lisani [16] résout l'extraction et propose une méthode complète (extraction, codage, comparaison) à laquelle manque seulement la phase finale de décision. Plus précisément, ses seuils sont multiples et ne sont pas automatiques. Nous allons proposer un cadre pour prendre cette décision. Nous calculons automatiquement un seuil de rejet/acceptation pour l'appariement de caractéristiques locales. Ce seuil dépend uniquement du contexte, c'est-à-dire de la forme cherchée elle-même et de la base dans laquelle s'effectue la recherche.

Notre plan est le suivant. Après la présente introduction faisant le point sur la reconnaissance de formes, nous rappelons l'algorithme de codage des images proposé par Lisani, sur lequel nous nous appuyons (partie 2). Nous expliquons ensuite notre méthode de décision rejet/acceptation pour la reconnaissance. Le nombre de fausses alarmes d'un appariement est une mesure de qualité (section 3). Enfin, nous présentons des résultats pratiques et discuterons s'ils valident bien l'approche proposée (section 4).

2. comparaison d'images par leurs formes

Ce qui est expliqué ici n'est pas nouveau, mais est essentiel pour la compréhension de la méthode de calcul du nombre de fausses alarmes.

Nous décrivons dans cette section les deux étapes d'extraction et de codage reprises de [16], que nous allons ensuite compléter pour permettre la prise de décision. Il est aisé de vérifier que le codage de morceaux de forme obtenu par cette méthode est :

– invariant par changement de contraste affine,

– invariant par similitude,

– robuste aux occlusions dans la mesure où il est un codage local de forme.

1. *Extraction et filtrage des lignes de niveau pour chaque image*. L'ensemble des lignes de niveau d'une image contient toute l'information de forme, dans le sens où le contour apparent de chaque objet visible est généralement bien suivi par l'union d'un très petit nombre de morceaux de lignes de niveau, et même très souvent par une seule ligne de niveau. Si nous considérons toutes les lignes de niveau, ou même un choix assez restreint, nous avons de plus une représentation complète car l'image peut être reconstruite à partir de ses lignes de niveau, et une image quantifiée assez fidèle peut être reconstruite à partir d'un choix très restreint de lignes de niveau, par un algorithme trivial. La représentation de l'image en lignes de niveau n'est pas invariante par changement d'*illumination* de la scène : de tels changements d'illumination peuvent la changer complètement, et aucun descripteur ne pourrait être invariant en ce sens-là. Mais les lignes de niveau sont invariantes par changement du *contraste* de l'image. La plupart d'entre elles n'a néanmoins pas d'intérêt sémantique : ce sont des lignes oscillant aléatoirement dans les zones homogènes de l'image (figure 1(b)). La grande quantité de lignes de niveau oblige à faire une sélection des lignes perceptuellement « importantes ». Pour sélectionner ces lignes, on peut utiliser la technique présentée par Desolneux *et al.* [9], qui définit les lignes significatives comme étant suffisamment contrastées et longues, au sens du principe de Helmholtz (figure 1(c)). On observe la formation de bords épais, constitués par un ensemble de lignes de niveau significatives incluses les unes dans les autres. Si on considère l'arbre d'inclusion des lignes de niveau de l'image (cet arbre est calculé de manière rapide au moyen de la *Fast Level Set Transform*, introduite par Monasse [20]) ces bords épais correspondent à des branches telles que chaque nœud n'a qu'un seul fils, et telles que le niveau de gris des courbes de toute la branche soit croissant, ou décroissant, avec la profondeur dans l'arbre. Afin de représenter les bords avec une seule ligne, on choisit dans chaque branche monotone la ligne de niveau la plus significative au sens contraste-longueur. Il s'agit d'un algorithme sans paramètre si on fixe comme il est raisonnable son nombre de fausses alarmes à 1. La figure 1(d) montre l'ensemble des lignes de niveau obtenues finalement, dites *lignes significatives maximales*. Ces lignes fournissent une représentation compacte des formes contenues dans l'image. Cette étape n'a d'autre objectif que de réduire le nombre de courbes à coder, ce qui accélère l'étape de décision.

Une fois les lignes extraites, nous leur appliquons le schéma rapide de Moisan [18] pour le lissage invariant par transformation affine [23] :

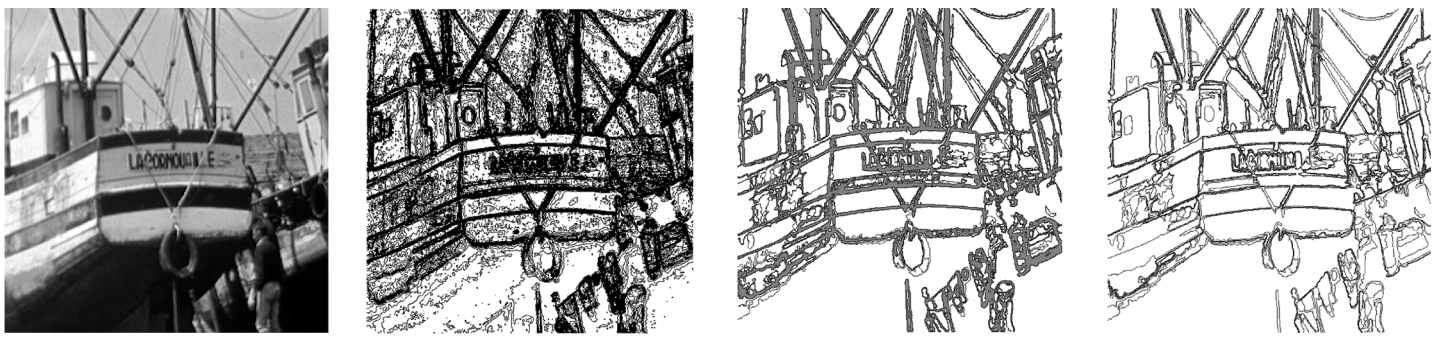
$$\frac{\partial x}{\partial t} = |\mathrm{Curv}(x)|^{\frac{1}{3}}\,\vec{n}(x),$$

où x est un point d'une ligne de niveau, $\mathrm{Curv}(x)$ la courbure et $\vec{n}(x)$ la normale à la courbe, orientée vers la concavité. Le but du lissage est uniquement à nouveau d'accélérer la méthode en réduisant le nombre de codes associé à chaque ligne de niveau. L'échelle du lissage est fixe, adaptée à l'élimination des effets d'*aliasing* (escaliers sur les bords) qui ont une taille de l'ordre du pixel. Ce lissage est donc sans paramètre.

2. *Codage des lignes de niveau et création du dictionnaire associé à l'image*. En choisissant des invariants selon le type de codage à effectuer, on définit des repères locaux pour l'ensemble des lignes de niveau issues de l'étape précédente. Ceci permet de définir des codes pour les morceaux de courbe, en prenant des morceaux à longueur normalisée fixe et en les échantillonnant régulièrement dans le repère normalisé. Le code correspondant à un morceau de courbe est l'ensemble des échantillons normalisés $\{(x_i, y_i), 1 \leqslant i \leqslant N\}$. La localité du codage assure une certaine robustesse vis-à-vis des occlusions. Ainsi, à chaque image on associe un dictionnaire qui contient tous les codes extraits de ses lignes de niveau significatives.

Voici l'algorithme de codage précis pour le cas similitude-invariant [15]. Le cas affine-invariant est également possible et est décrit dans [16].

Pour coder une ligne de niveau $\mathcal{L}$, pour chaque point plat, point d'inflexion, et chaque couple de points en lesquels la même droite est tangente à la courbe (voir figure 2) :



(a) **(b)** **(c)** **(d)**

Figure 1. – Extraction des lignes de niveau significatives. (a) image originale, (b) lignes tous les 10 niveaux de gris (5479 lignes), (c) lignes de niveau significatives (4342 lignes), (d) lignes significatives maximales (296 lignes).

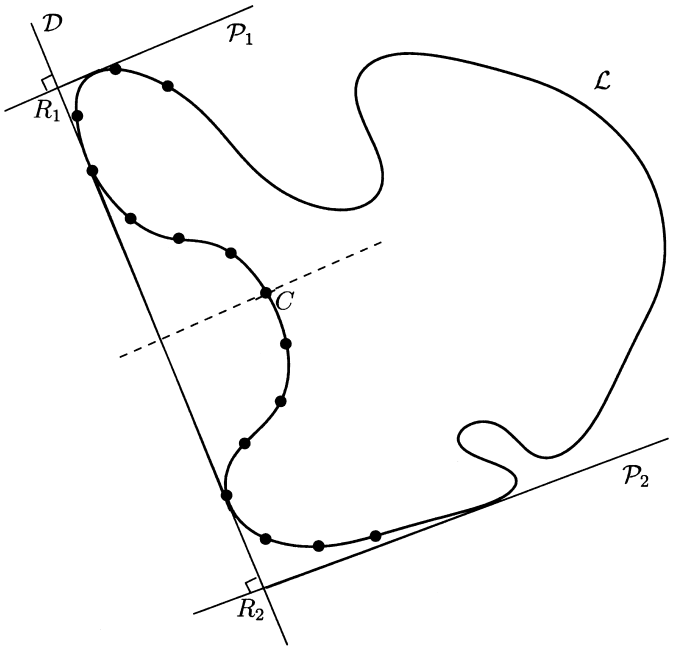


Figure 2. – Codage similitude invariant d'un morceau de ligne de niveau. Ici $N = 15$ et $F = 2$; la direction utilisée pour le codage est celle d'une bitangente. Les coordonnées des points marqués dans le repère orthonormé s'appuyant sur $[R_1, R_2]$ forment le code d'un morceau de courbe.

1. considérer la tangente $\mathcal{D}$ en ce point ou couple de points ;
2. construire les deux premières perpendiculaires $\mathcal{P}_1$ et $\mathcal{P}_2$ à $\mathcal{D}$ qui sont tangentes à $\mathcal{L}$ (parcours de $\mathcal{L}$ dans un sens pour $\mathcal{P}_1$, dans l'autre sens pour $\mathcal{P}_2$) ;
3. calculer les coordonnées des points R_1 et R_2, intersection de $\mathcal{D}$ avec respectivement $\mathcal{P}_1$ et $\mathcal{P}_2$;
4. repérer N points équirépartis sur un arc de $\mathcal{L}$, de longueur $F \cdot \|R_1R_2\|$, centré sur le point d'intersection C de la médiatrice du segment $[R_1R_2]$ avec $\mathcal{L}$.

La donnée des deux points R_1 et R_2 permet de construire un repère invariant par similitude ; les coordonnées des N points dans ce nouveau repère fournissent la description invariante d'un morceau de la ligne de niveau. Il se trouve que le choix $N = 45$ et $F = 5$ fournit une description suffisamment locale et précise des morceaux, ce sont ces valeurs qui sont utilisées dans les expériences de la partie 4. Cette méthode permet de bien coder toutes les formes suffisamment non convexes. Par exemple, la lettre « P » dans le logo de la figure 7 n'est pas codée, ni le point. Néanmoins ce n'est pas notre objet ici de décrire une méthode complète, mais seulement d'expliquer la méthode d'acceptation-rejet.

La méthode originale de Lisani, s'inspirant du *Geometric Hashing*, utilise ces codes afin d'identifier des pré-appariements qui sont ensuite vérifiés par recalage du code candidat et de la requête. Nous n'utiliserons pas ceci dans la suite.

3. introduction d'un seuil de rejet/acceptation

Pour aboutir à une méthode plus robuste, nous voudrions éliminer les paramètres de l'étape d'appariement, en particulier les seuils d'appariement dont le choix n'est pas du tout intuitif.

Ce choix, au lieu d'être laissé à l'utilisateur, doit être automatique ; il sera déduit de la distribution des « distances » entre les codes de requête et ceux de la base.

Chaque code extrait est en fait un N-uplet de couples $\{(x_i, y_i), 1 \leqslant i \leqslant N\}$. L'idée est alors d'établir une mesure de proximité entre codes, dépendante du contexte. Ce contexte est donné par une base de formes issues d'une ou plusieurs images. La méthode est relative à cette base, et les résultats de la reconnaissance changent si on change la base.

Dans un premier paragraphe, nous bâtissons un cadre général pour la reconnaissance de formes. Les « codes » auxquels nous nous référerons ne seront pas ces N-uplets, mais des codes idéaux dont les composantes présentent la particularité d'être statistiquement indépendantes. Nous expliquons dans un second paragraphe comment transformer nos N-uplets de points en de tels codes, et comment interpréter l'appariement des codes en termes de reconnaissance de formes.

3.1. significativité des appariements de codes normalisés

En premier lieu, nous construisons un modèle statistique empirique de la base de formes. Les appariements corrects seront détectés *a contrario* comme des événements rares de ce modèle. Ce cadre de détection a été récemment appliqué à la détection d'alignements [8] ou de bords contrastés [9] par A. Desolneux *et al.*, à la détection de points de fuite par A. Almansa *et al.* [1], et par F. Cao à la détection de bonnes continuations [5]. Il dérive d'un principe perceptuel (le principe de Helmholtz selon la terminologie de A. Desolneux) qui énonce que si le nombre d'apparitions « par hasard » d'un événement est très faible, alors cet événement est particulièrement *significatif*. Cette technique présente l'avantage de ne nécessiter qu'un seul paramètre pour contrôler la détection : le nombre de fausses alarmes. Il s'agit d'une quantité qui a une signification précise et qui peut toujours être laissée à un utilisateur non expert.

Toutefois, il y a une nouveauté dans la méthode que nous allons proposer. Les méthodes mentionnées ont un modèle de fond (*background model*) uniforme aléatoire. Si nous appliquions à la lettre la méthode de Desolneux *et al.* nous modéliserions les formes comme des trajectoires browniennes. Or, pour avoir un seuil correct d'acceptation / rejet, il faut donner un modèle statistique le plus fin et précis possible de la base de formes réelle.

Cela veut aussi dire que la reconnaissance de forme est une notion relative à une base. Cette relativité est indispensable : il n'y a pas d'évaluation possible pour les « ressemblances casuelles » sans prise en compte de la base de formes dans laquelle cette ressemblance peut se produire.

Supposons que le problème soit de décider si le code d'une forme correspond au code d'une autre forme dans une base de N_B codes. Nous supposons que les codes des formes sont des listes de n caractéristiques, chacune d'entre elles appartenant à un espace métrique (E_i, d_i), $1 \leqslant i \leqslant n$.

Soit $X = (x_1, \ldots, x_n)$ un code de requête, et soit $Y = (y_1, \ldots, y_n)$ un élément du produit cartésien $E_1 \times \ldots \times E_n$. Étant donnés n réels positifs $\delta_1, \ldots, \delta_n$, nous disons que X et Y sont $(\delta_1, \ldots, \delta_n)$-proches si et seulement si :

$$\forall i \in \{1, \ldots, n\},\ d_i(x_i, y_i) \leqslant \delta_i.$$

Deux codes se correspondent s'ils sont $(\delta_1, \ldots, \delta_n)$-proches avec $\delta_1, \ldots, \delta_n$ suffisamment petits. On voit qu'il est obligatoire de fixer un seuil pour les δ_i.

Supposons, hypothèse cruciale, que les caractéristiques de forme en question soient statistiquement indépendantes. Il est alors possible de calculer la probabilité, notée $\mathcal{P}(X, \delta_1, \ldots, \delta_n)$ dans ce qui suit, pour qu'un code Y soit $(\delta_1, \ldots, \delta_n)$-proche d'un code donné X :

$$\Pr(Y \text{ t.q. } Y \text{est } (\delta_1, \ldots, \delta_n)\text{-proche de} X) = \prod_{i=1}^{n} \Pr(y \in E_i \text{ t.q. } d_i(y, x_i) \leqslant \delta_i). \quad (1)$$

La construction des caractéristiques indépendantes doit dépendre de la base de formes et chaque terme du produit précédent est alors estimé de manière empirique sur la base : pour chaque i, nous calculons la fonction de distribution de $d_i(z, x_i)$, où z parcourt l'ensemble des $i^{\text{ème}}$ caractéristiques des codes de la base. De cette manière nous calculons $\mathcal{P}(X, \delta_1, \ldots, \delta_n)$.

Définition 1 (appariement ε-significatif). *Étant donné $X = (x_1, \ldots, x_n)$, nous dirons qu'un code $Y = (y_1, \ldots, y_n)$ est un appariement ε-significatif de X si on a :*

$$N_B \cdot \left(\max_{1 \leqslant i \leqslant n} \Pr(y \in E_i \text{ t.q. } d_i(y, x_i) \leqslant \delta_i) \right)^n \leqslant \varepsilon,$$

où $\forall i \in \{1 \ldots n\}, \delta_i = d_i(y_i, x_i)$.

Remarquons que la condition de la définition précédente est équivalente à :

$$\forall i \in \{1 \ldots n\}, \Pr(y \in E_i \text{ t.q. } d_i(y, x_i) \leqslant \delta_i) \leqslant \left(\frac{\varepsilon}{N_B} \right)^{\frac{1}{n}}. \quad (2)$$

On impose une borne uniforme sur les probabilités relatives à chaque caractéristique car dans notre cas il n'y a pas de raison de les différencier.

Les fonctions $d \mapsto \Pr(y \in E_i \text{ t.q. } d_i(y, x_i) \leqslant d)$ étant croissantes, elles sont pseudo-inversibles, et donc il existe des réels δ_i^* maximaux (qui dépendent de X et ε) tels que si $\delta_i < \delta_i^*$, alors l'inégalité 2 est vérifiée.

D'où la proposition :

Proposition 1. *Un code $Y = (y_1, \ldots, y_n)$ est un appariement ε-significatif de $X = (x_1, \ldots, x_n)$ si et seulement si : Y est $(\delta_1, \ldots, \delta_n)$-proche de X, où les δ_i vérifient :*

$$\forall i \in \{1 \ldots n\}, \delta_i < \delta_i^*.$$

où $\forall i \in \{1 \ldots n\}$,

$$\delta_i^* = \max \left\{ d > 0, \Pr(y \in E_i \text{ t.q. } d_i(y, x_i) \leqslant d) \leqslant \left(\frac{\varepsilon}{N_B} \right)^{\frac{1}{n}} \right\}.$$

On peut remarquer que les probabilités empiriques prennent en compte la « rareté » ou la « banalité » dans la base d'un appariement potentiel. En effet, les seuils δ_i^* sont moins restrictifs dans le premier cas et plus stricts dans l'autre cas.

La proposition suivante justifie la définition de la significativité et montre que le nombre de fausses détections est contrôlé par ε. Ceci est une manière de contrôler les détections beaucoup plus intuitive que de se contenter de régler les seuils de distance δ_i^* à la main pour chaque requête.

Proposition 2. *L'espérance du nombre d'appariements ε-significatifs sur l'ensemble de tous les codes de la base est inférieur à ε.*

Preuve : Notons $Y_j = (y_1^j, \ldots, y_n^j)$ $(1 \leqslant j \leqslant N_B)$ les codes possibles, et χ_j la fonction indicatrice de l'événement e_j : « Y_j est apparié significativement avec X ». Soit $R = \sum_{j=1}^{N_B} \chi_j$ la variable aléatoire représentant le nombre d'appariements ε-significatifs de X. L'espérance de R est $E(R) = \sum_{j=1}^{N_B} E(\chi_j)$. D'après la définition 1, $E(R) = \sum_{j=1}^{N_B} \mathcal{P}(X, d_1(y, x_1^j), \ldots, d_n(y, x_n^j))$, donc $E(R) \leqslant \sum_{j=1}^{N_B} \varepsilon \cdot N_B^{-1}$, ce qui entraîne $E(R) \leqslant \varepsilon$.

Remarque : Le point clé est que nous contrôlons l'espérance de R. Comme la dépendance entre les événements e_j est inconnue, nous ne sommes pas capables d'estimer la loi de probabilité de R. Néanmoins, la linéarité permet de calculer l'espérance. Certains auteurs ayant essayé de traiter ce problème [21, 12, 13] supposaient arbitrairement que les événements sont indépendants.

Pour finir avec ces définitions, nous donnons une mesure de la qualité d'un appariement.

Définition 2 (nombre de fausses alarmes). *Étant donné un code X et des distances $\delta_1,\ldots,\delta_n > 0$, nous appelons nombre de fausses alarmes d'un appariement avec X à la distance $\delta_1,\ldots,\delta_n$ le nombre :*

$$NFA(X, \delta_1,\ldots,\delta_n) = N_B \cdot \mathcal{P}(X, \delta_1,\ldots,\delta_n).$$

Le nombre de fausses alarmes d'un code X à des distances $\delta_1,\ldots,\delta_n$ données est une estimation du nombre de codes $\delta_1,\ldots,\delta_n$-proches de X dans la base.

Le cadre que nous venons de présenter permet de détecter des appariements en contrôlant le nombre d'objets qui sont appariés uniquement « par hasard ».

Si nous voulons que ces appariements par hasard entre un code de requête et un code de la base n'arrivent en moyenne qu'une seule fois, nous fixons tout simplement $\varepsilon = 1$. Si la requête est faite de N_Q codes d'importance égale, et si nous voulons détecter en moyenne un seul appariement par hasard *sur tous les codes de requête*, nous fixons encore $\varepsilon = 1$ après avoir remplacé N_B par $N_B \cdot N_Q$ dans la définition 1 (dans ce cas, la proposition 2 se démontre de la même manière).

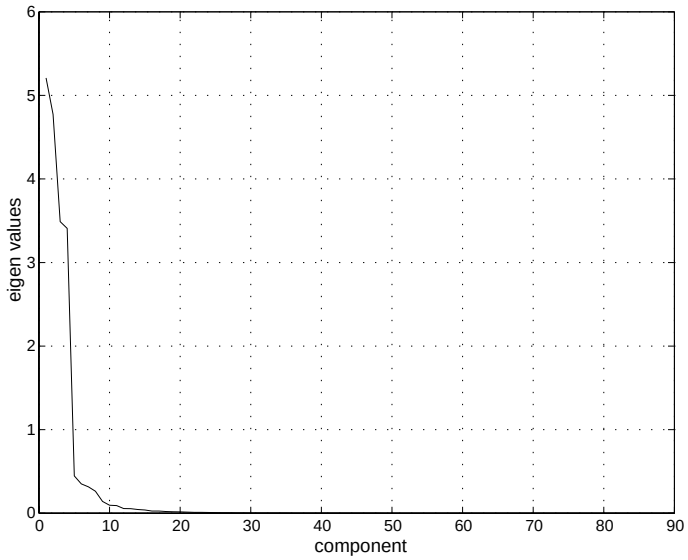


Figure 3. – Exemple des 90 valeurs propres de l'analyse en composantes principales, classées par ordre décroissant (il s'agit des valeurs pour l'expérience de la section 4.)

3.2. modélisation

L'objet de cette section est de décrire comment un code peut être traité pour entrer dans le cadre de la section 3.1.

Les codes de la base sont déterminés selon l'algorithme développé dans la partie 2. Ce sont des listes de coordonnées de 45 points sur des parties de lignes de niveau.

Voici comment on associe à chaque code une liste de caractéristiques. Tout d'abord on considère que chaque code de 45 points est un vecteur de $\mathbb{R}^{90}$. L'ensemble des codes de la base est centré en son barycentre dans $\mathbb{R}^{90}$. On procède ensuite à son analyse en composantes principales, pour extraire des composantes si possible indépendantes et *a priori* décorrélées: un repère orthonormé $\{e_1,\ldots,e_{90}\}$ est identifié, les e_i étant ordonnés selon les variances décroissantes (*cf.* figure 3). On projette ensuite chaque code X de la base sur l'espace vectoriel engendré par les M premiers vecteurs $\{e_1,\ldots,e_M\}$. Les coordonnées dans ce repère composent la liste de caractéristiques $(x_1,\ldots,x_M)$. De la même façon les codes de requête sont projetés dans ce sous-espace. Il est clair que plus M est grand, plus la description de la forme est complète. Pour reprendre les notations de la partie 3.1, ici : $(E_i,d_i) = (\mathbb{R},|\cdot|)$. La figure 3 montre que le choix $M = 10$ est raisonnable.

Notre modèle *a contrario* suppose que les caractéristiques sont indépendantes. Bien qu'on ne puisse prétendre que les composantes résultant de l'analyse en composantes principales soient indépendantes, elles sont au moins décorrélées. Nous aurions pu choisir comme caractéristiques les coordonnées des points formant les codes. Mais ces caractéristiques ne sont pas du tout indépendantes, ce qui empêche d'évaluer correctement la formule 1.

D'après le modèle, l'appariement de deux codes (dans le sens de la section 3.1) est le résultat d'un événement rare. Afin d'établir le lien entre les appariements détectés et la reconnaissance de formes, nous devons clairement expliquer les appariements.

Dès que le nombre de détections dépasse largement ε, on a affaire à ce qu'on appelle *Détections Identifiantes* (DI) et *Détections Non-Identifiantes* (DNI). Ces détections doivent pour la plupart être expliquées comme des violations de l'hypothèse d'indépendance. Cette violation peut être due à deux facteurs.

a) Des codes similaires sont appariés car ils ont été extraits d'objets similaires (ou identiques). Nous appelons les appariements de ce type *Détections Identifiantes* et c'est bien l'objet de notre méthode de les trouver !

b) De nombreuses formes dans les images dérivent d'objets naturels ou créés par l'homme ayant une structure commune. Par exemple, de nombreux objets présentent du parallélisme, ou une constance de largeur. Ceci peut entraîner des déviations par rapport au modèle d'indépendance. Ces *Détections Non Identifiantes* peuvent aussi être dues à l'incomplétude du codage par les premières composantes de l'analyse en composantes principales. Enfin, un nombre de l'ordre de ε des DNI peuvent être ce que nous appellerions proprement les fausses détections. Les *fausses détections* sont seulement détectées parce que leurs caractéristiques sont proches « par hasard ». Selon la proposition 2, il devrait y avoir, en moyenne, au plus ε détections de cette nature. Il est bien-sûr souvent difficile de les distinguer des DNI.

Même si elles sont sémantiquement différentes, les deux types de détections (DI et DNI) ne peuvent être distinguées de notre point de vue. Notre méthode ne peut pas contrôler le nombre de Détections Non-Identifiantes, car elles ne sont pas toutes des Fausses Détections. Si nous prenions deux codes provoquant une DNI hors de leur contexte, nous considérerions leur appariement comme correct (voir figure 6 dans la section 4.2).

4. mise en œuvre pratique

Dans toutes les expériences présentées dans cette section nous utilisons le codage invariant par similitude. Les morceaux de courbes codés sont ceux dont la longueur d'arc normalisée est 5. Les codes sont constitués de 45 points issus d'un échantillonnage régulier des morceaux de courbes normalisés.

4.1. vérification du modèle

Afin de montrer la validité de la méthode sous l'hypothèse d'indépendance, nous procédons à l'expérience suivante : 100 000 codes sont générés aléatoirement selon un modèle de marche aléatoire à pas constant (distances entre deux points consécutifs constante, angles répartis uniformément) puis on cherche un code parmi ces 100 000. Des codes générés aléatoirement vérifient l'indépendance, contrairement à des codes issus de lignes de niveau extraites d'images réelles. En effet, ces derniers sont contraints à ne pas s'intersecter et peuvent présenter des causalités communes (parallélisme, constance de largeur, *etc.*), ce qui biaise l'estimation des probabilités. Les tableaux suivants montrent le nombre de détections pour 3 courbes différentes, qui sont des réalisations du même processus ayant engendré les courbes de la base, selon différentes valeurs de la significativité ε et du nombre de composantes. Le nombre de détections est bien de l'ordre de ε, comme prévu par la théorie.

ε	1	10	100	1000
20 composantes	0	9	77	823
10 composantes	1	10	88	939
5 composantes	1	11	127	1006

Courbe 1.

ε	1	10	100	1000
20 composantes	2	10	83	895
10 composantes	0	6	88	944
5 composantes	1	9	95	964

Courbe 2.

ε	1	10	100	1000
20 composantes	0	8	96	822
10 composantes	0	10	93	916
5 composantes	1	10	94	954

Courbe 3.

L'expérience prouve *a posteriori* que l'analyse en composantes principales a donné des variables indépendantes, et donc que les calculs menés sous cette hypothèse sont valides. En effet, la première ligne annonce le nombre de détections attendues sous hypothèse d'indépendance des composantes. Le tableau montre que, quel que soit le nombre de composantes choisi, le nombre de détections observé est très proche du nombre prédit. Nos raisonnements conduisent à prédire un nombre de détections inférieur ou égal à ε. On constate que la prédiction est en fait plus précise que cela.

4.2. détection de morceaux de lignes de niveau

L'expérience suivante permet de vérifier les invariants de la méthode (contraste, occlusion, similitude). Nous avons cherché les lignes d'une image parmi celles d'une image de référence la contenant (voir figure 4). Les deux images sont deux prises de vues différentes du même tableau. Le nombre de composantes de l'analyse en composantes principales considérées est de 10.

L'image 5 montre le résultat de cette expérience. Parmi les 975 codes de l'image de requête, 26 trouvent au moins un appariement dans les codes de l'image de référence, qui comporte 38700 codes. En tout 53 appariements sont détectés (ce qui signifie que certains codes de requête correspondent à plusieurs codes de la base, qui sont en fait de petites variations du même code). Les cinq détections non identifiantes sont montrées dans la figure 6.

4.3. application à la recherche de logos dans une base de publicités

Nous disposons d'une part d'un logo (*cf.* figure 7), qui sera l'image de requête, et d'autre part d'une base d'images constituée de 21 publicités, comportant toute sorte de structure ou texture (*cf.* figure 8). Le nombre de codes correspondant est 90000. Le codage est basé sur les 10 composantes de plus forte variance dans l'analyse en composantes principales.

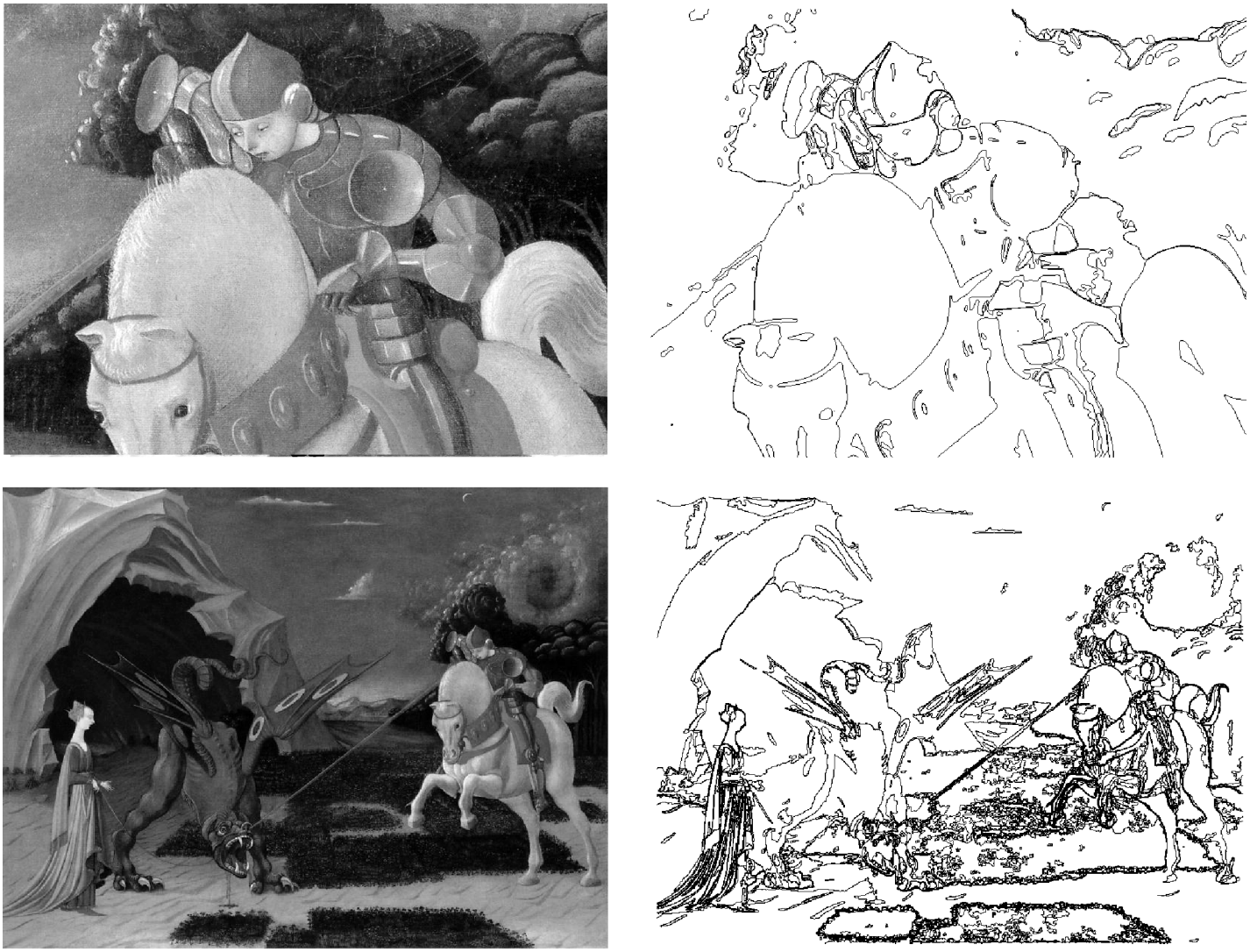


Figure 4. – En haut : image de requête et lignes de niveau extraites. En bas : image dans laquelle s'effectue la recherche, et ses lignes de niveau. Ces images sont deux prises de vues différentes du même tableau (*Saint-Georges et le dragon*, Paolo Uccello). En particulier, le contraste n'est pas le même sur les deux images.

Figure 5. – Mise en évidence des codes détectés sur l'image de référence, pour un nombre de fausses détections égal à 1. Des détections non-identifiantes peuvent être vues dans la partie gauche.

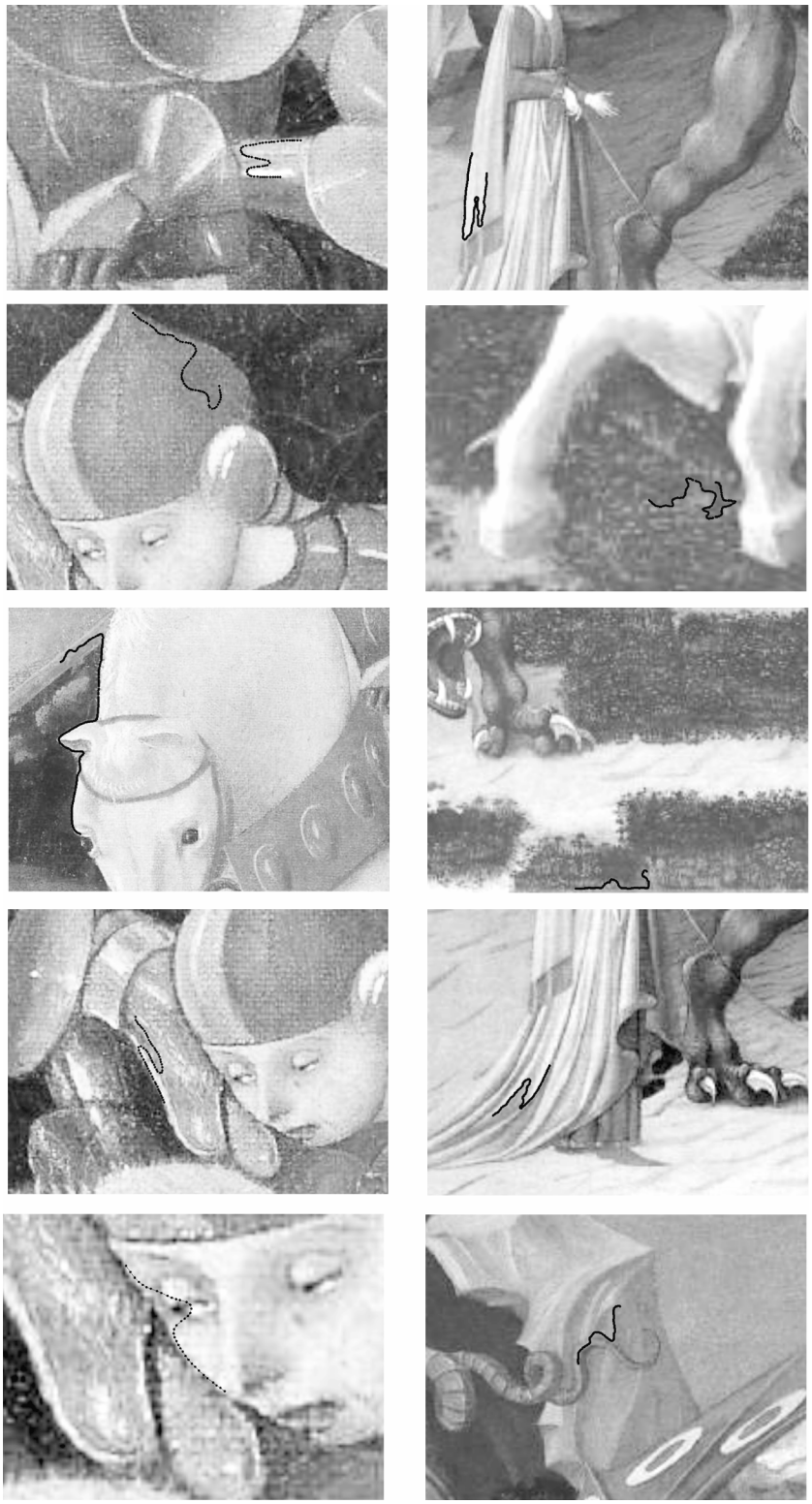


Figure 6. – Les cinq Détections Non Identifiantes dans l'image requête (à gauche) et dans l'image de référence (à droite). Seule la deuxième détection, et éventuellement la cinquième, peuvent vraiment être considérées comme des fausses détections (ici $\varepsilon = 1$).

Les résultats sont donnés figures 9 à 11 (à gauche, codes du logo de requête ; à droite, codes en correspondance dans les images de la base). Les figures 9 et 10 montrent toutes les images de la base pour lesquelles une détection à été faite quand $\varepsilon = 1$. Il y a à $\varepsilon = 1$ un mélange de NID et ID. Toutefois, aucune NID n'a un $NFA \leqslant 10^{-1}$, comme le prouve la planche de la figure 11. Par contre, le nombre de fausses alarmes des DI descend jusqu'à $\varepsilon = 10^{-8}$. On peut interpréter ce résultat de la manière suivante : une telle détection, si la base d'images d'apprentissage est représentative, se maintiendrait encore dans une base d'images comportant 10^7 images et aurait alors un NFA de 10^{-2}. Le NFA permet de quantifier la taille de la base où une reconnaissance est possible.

Figure 7. – Requête : le logo « puma ». À droite, lignes de niveau significatives extraites.

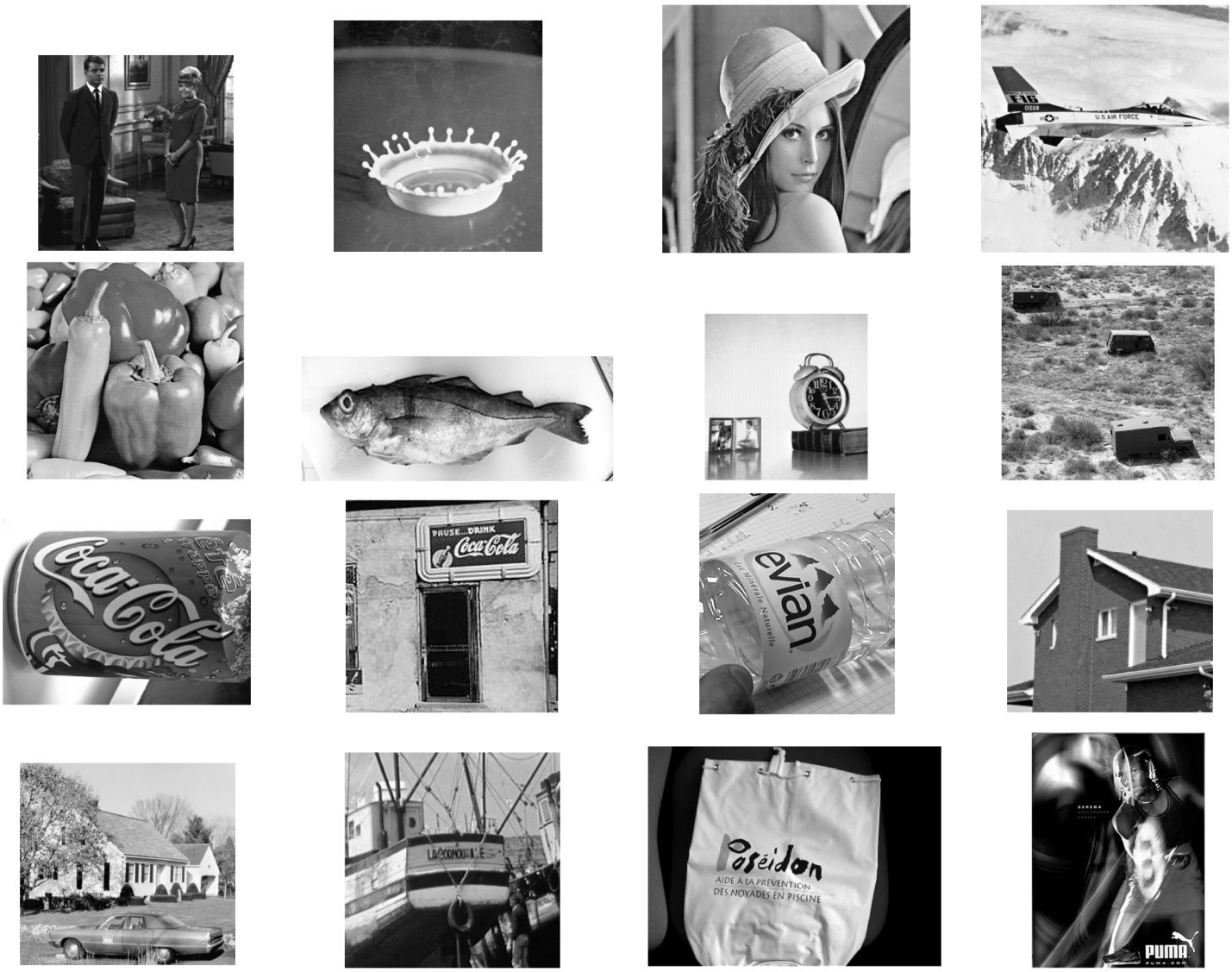


Figure 8. – La base d'images, correspondant aux 90000 codes extraits.

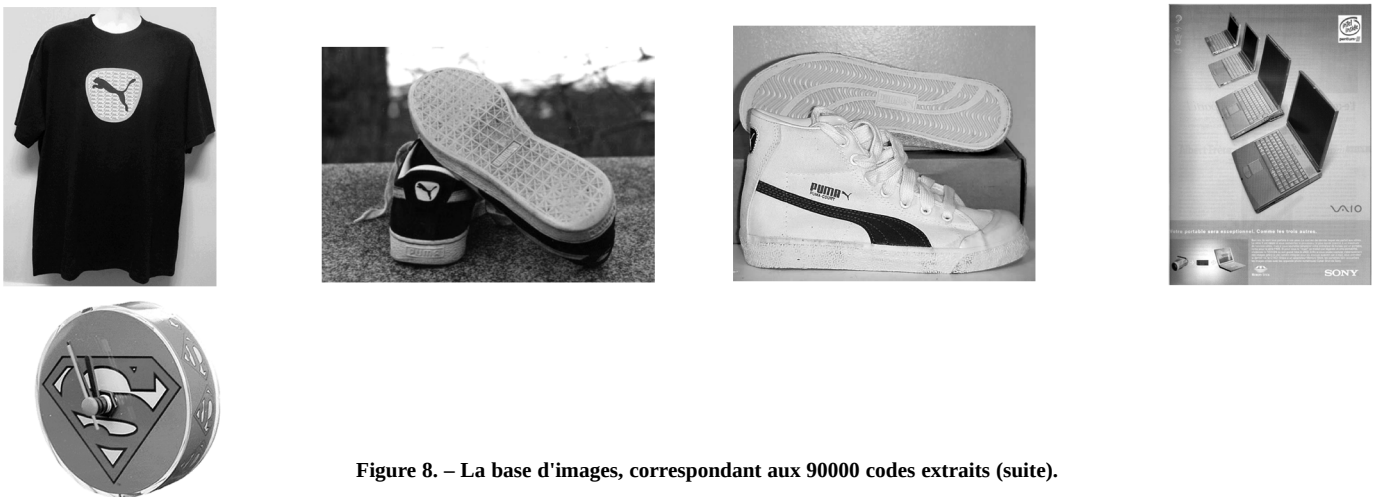


Figure 8. – La base d'images, correspondant aux 90000 codes extraits (suite).

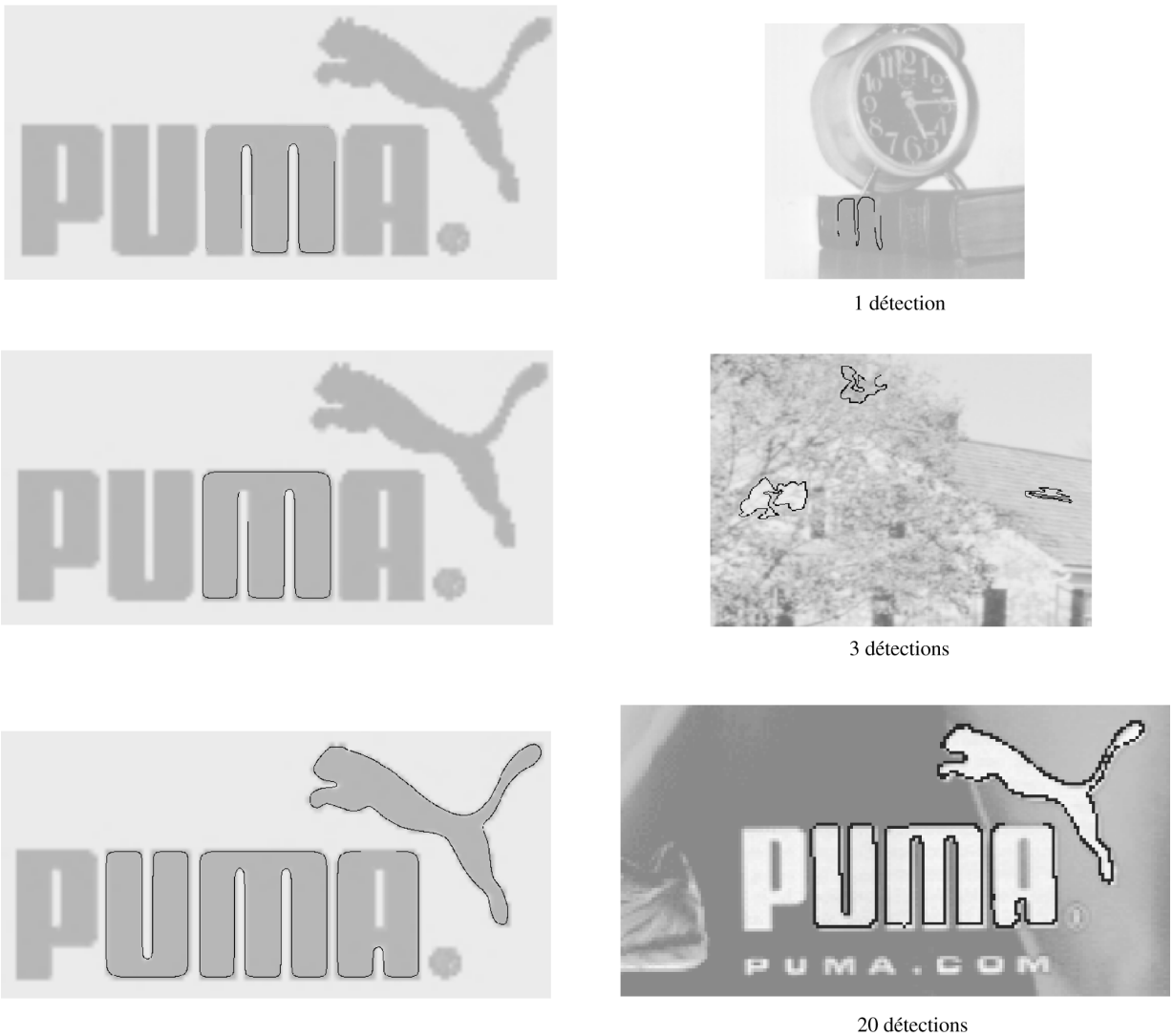
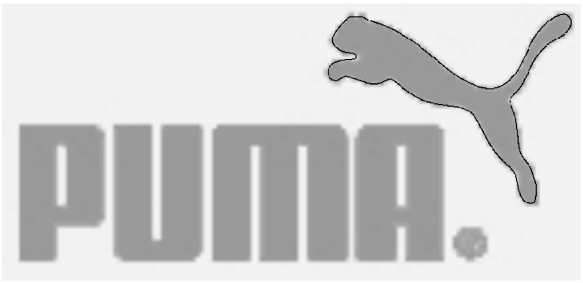
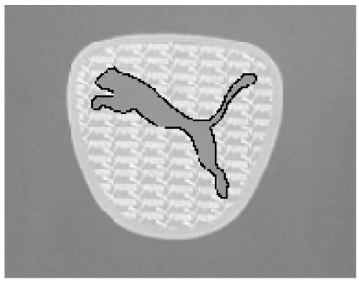
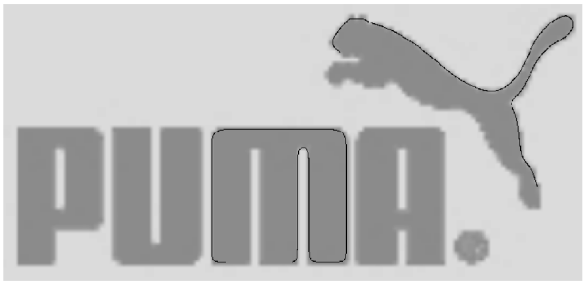


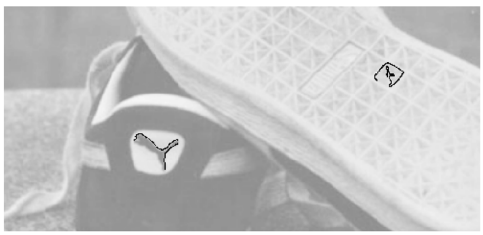
Figure 9. – Détection de logo. $\varepsilon = 1$.



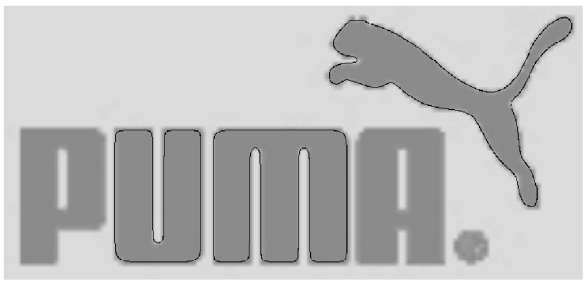


7 détections





3 détections



14 détections

Figure 10. – Détection de logo. $\varepsilon = 1$**.**

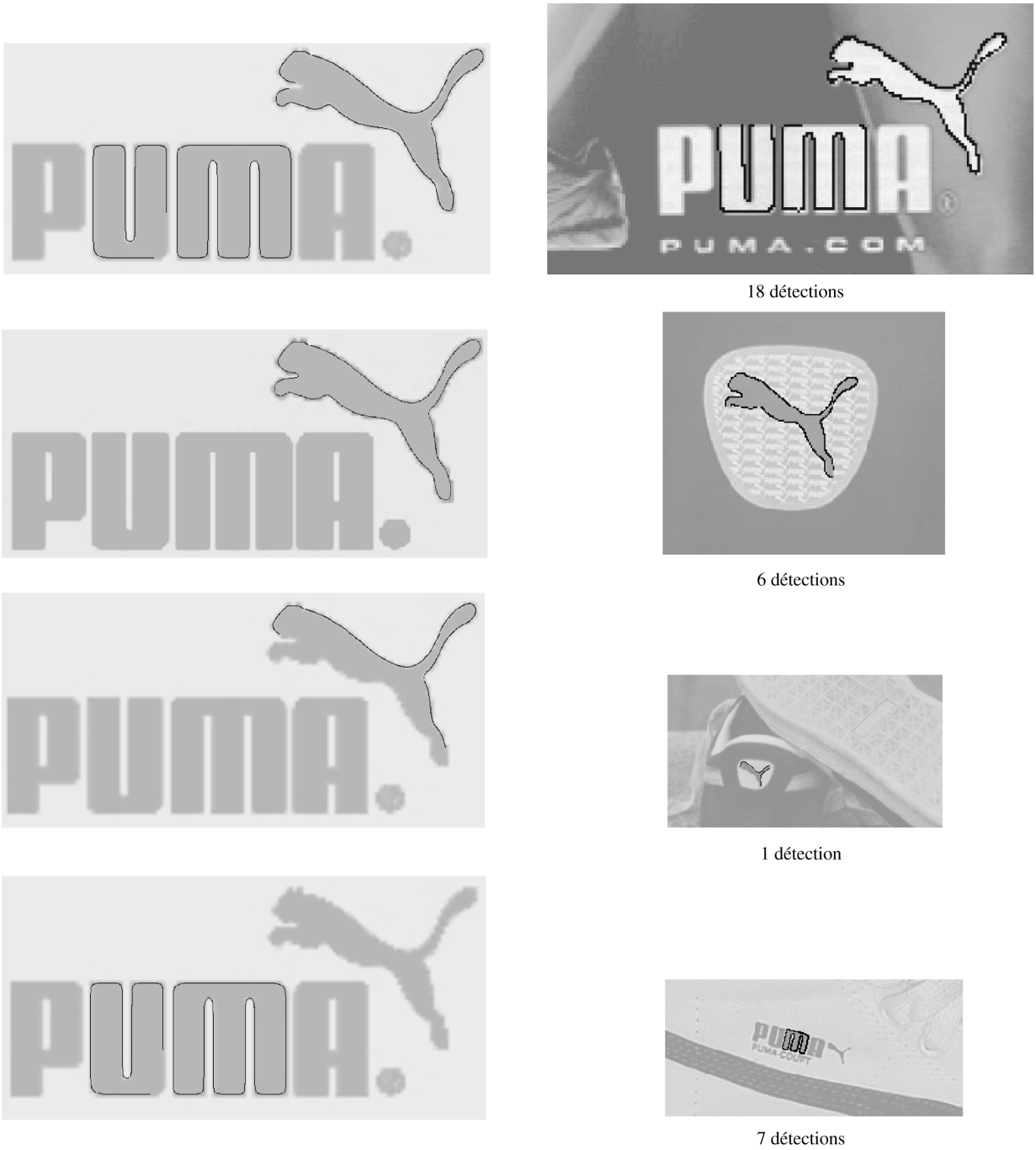


Figure 11. – Détection de logo. $\varepsilon = 10^{-1}$ **. Des détections identifiantes se maintiennent jusqu'à, de haut en bas,** $\varepsilon = 10^{-8}$**,** $\varepsilon = 10^{-4}$**,** $\varepsilon = 10^{-2}$ **et** $\varepsilon = 10^{-2}$**.**

5. conclusion

Dans cet article, nous avons montré la possibilité de définir des seuils absolus de reconnaissance de formes, relativement à une base de formes. Ces seuils sont robustes au sens où les détections identifiantes se distinguent par un nombre de fausses alarmes très inférieur à celui des détections non-identifiantes. Nous avons vérifié qu'il suffit, quelle que soit la base de référence et la forme à reconnaître de fixer le seuil entre 10^{-1} et 10^{-2} pour s'assurer d'une reconnaissance sans « faux positif ».

Nous avons conçu les formes comme des réalisations d'une variable aléatoire empirique observable. Cette variable aléatoire vectorielle a été décomposée en composantes principales dont l'expérience a montré qu'elles étaient proches de l'indépendance. Néanmoins cette décomposition n'est pas complète car on ne retient pas autant de composantes que la dimension de l'ensemble de formes. En conséquence, deux formes risquent d'avoir des représentations semblables sans pour autant être à une distance de Hausdorff petite. Par ailleurs il est impossible de retenir toutes les composantes de l'ACP, car la ressemblance des composantes à variance faible devient aléatoire et l'exiger conduit à perdre des détections.

Aussi, le problème reste ouvert de trouver pour une variable aléatoire empirique comme celle donnée par l'ensemble des codes extraits une description en composantes indépendantes et qui soit aussi complète vis-à-vis de la distance de Hausdorff.

La méthode présentée n'est pas complète : il manque la prise en compte du fait que plusieurs parties d'une forme donnent des codes qui se correspondent de manière spatialement cohérente quand la requête (un logo par exemple) est présente. L'autre problème ouvert est donc le calcul du nombre de fausses alarmes pour un événement plus sophistiqué prenant en compte cette coïncidence supplémentaire. Nous avons concentré cette étude sur le rejet des « faux positifs » et laissé de coté le problème des « faux négatifs », à savoir tous les cas où une forme recherchée n'est pas trouvée bien que présente. Selon les expériences communiquées par Andrés Almansa, une grande partie des échecs en détection (les faux négatifs) sont dus à des problèmes d'extraction, et non pas à la méthode de codage de Lisani elle-même. Il est clair que la partie détection de frontières significatives est améliorable et nous comptons nous en occuper dans un travail futur.

remerciements

Ce travail a été soutenu par le projet ISII du RNRT, Ministère de la Recherche et de la Technologie, l'Office of Naval Research, grant N00014-97-1-0839, le Centre National d'Études Spatiales (contrat Recherche d'invariants). Les auteurs remercient Laurent Younès pour ses suggestions ayant conduit à l'introduction de l'analyse en composantes principales, Frédéric Cao et Andrés Almansa pour leurs remarques fructueuses, ainsi que les rapporteurs anonymes pour leur lecture minutieuse qui a permis de nombreuses améliorations.

BIBLIOGRAPHIE

[1] A. Almansa, A. Desolneux, S. Vamech, « Vanishing point detection without any a priori information » , *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, #4, 2003, p. 502-507.

[2] H. Alt, L. J. Guibas, « Discrete geometric shapes: Matching, interpolation, and approximation. A survey », Technical Report B 96-11, Universität Berlin, 1996.

[3] L. Alvarez, L. Mazorra, F. Santana, « Geometric invariant shape representations using morphological multiscale analysis adn applications to shape representation », *Journal of Mathematical Imaging and Vision*, Vol. 18, #2, 2002.

[4] D. H. Ballard, « Generalized Hough transfom to detect arbitrary patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, #2, 1981, p. 111-122.

[5] F. Cao, « Good continuations in digital image level lines », To appear in *Proceeding of ICCV 2003*.

[6] C. Carson, S. Belongie, H. Greenspan, J. Malik, « Blobworld: Image segmentation using expecation-maximization and its application to image querying », In *Third International Conference on Visual Information Systems*, 1999.

[7] T. Cohignac, « Reconnaissance de formes planes », PhD thesis, Ceremade, Université Paris IX Dauphine, 1994.

[8] A. Desolneux, L. Moisan, J.-M. Morel, « Meaningful alignments », *International Journal of Computer Vision*, Vol. 40, #1, 2000, p. 7-23.

[9] A. Desolneux, L. Moisan, J.-M. Morel, « Edge detection by Helmholtz principle », *Journal of Mathematical Imaging and Vision*, Vol. 14, #3, 2001, p. 271-284.

[10] S.A. Dudani, K.J. Breeding, R.B. McGhee, « Aircraft identification by moment invariants », *IEEE Transactions on Computers*, Vol. 26, #1, 1977, p. 39-46.

[11] Y. Gdalyahu, D. Weinshall, « Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, #12, 1999, p. 1312-1328.

[12] W.E.L. Grimson, D.P. Huttenlocher, « On the verification of hypothesized matches in model-based recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, #12, 1991, p. 1201-1213.

[13] D.P. Huttenlocher, E.W. Jaquith, « Computing visual correspondence: incorporating the probability of a false match », In *Proceedings of the Fifth International Conference on Computer Vision*, 1995, p. 572-594.

[14] Y. Lamdan, H. J. Wolfson, « Geometric hashing: a general and efficient model-based recognition scheme », In *2nd International Conference on Computer Vision*, 1988, p. 238-249.

[15] J.-L. Lisani, « Shape Based Automatic Images Comparison » , Thèse de doctorat, Université Paris 9 Dauphine, France, 2001.

[16] J.-L. Lisani, L. Moisan, J.-M. Morel, P. Monasse, « On the theory of planar shape », *SIAM Multiscale Modeling and Simulation*, Vol. 1, #1, 2003, p. 1-24.

[17] S. Loncarnic, « A survey of shape analysis techniques », *Pattern Recognition*, Vol. 31, #8, 1998, p. 983-1001.

[18] L. Moisan, « Affine plane curve evolution: A fully consistent scheme » , *IEEE Transactions on Image Processing*, Vol. 7, #3, 1998, p. 411-420.

[19] F. Mokhtarian, A. K. Mackworth, « A theory of multiscale, curvature-based shape representation for planar curves », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, #8, 1992, p. 789-805.

[20] P. Monasse, « Représentation morphologique d'images numériques et application au recalage », Thèse de doctorat, Université Paris 9 Dauphine, France, 2000.

[21] C. Olson, D.P. Huttenlocher, « Automatic target recognition by matching oriented edge pixels », *IEEE Transactions on Image Processing*, Vol. 6, #12, 1997, p.103-113.

[22] C.A. Rothwell, « Object Recognition Through Invariant Indexing », Oxford Science Publications, 1995.

[23] G. Sapiro, A. Tannenbaum, « Affine invariant scale-space », *International Journal of Computer Vision*, Vol. 11, #1, 1993, p. 25-44.

[24] C. Schmid, « A structured probabilistic model for recognition », In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, Vol. 2, 1999, p. 485-490.

[25] C. Schmid, R. Mohr, « Local greyvalue invariants for image retrievals », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, #5, 1997, p. 530-535.

[26] C. G. Small, « The Statistical Theory of Shapes », Springer, 1996.

[27] C.V. Stewart, « MINPRAN: a new robust estimator for computer vision », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, #10, 1995, p. 925-938.

[28] R. Veltkamp, M. Hagedoorn. « State-of-the-art in shape matching », Technical Report UU-CS-1999-27, Utrecht University, The Netherlands, 1999.

[29] R. C. Veltkamp, « Shape matching: similarity measures and algorithms », In *Proceedings of Shape Modelling International*, 2001, p. 188-197.

[30] A. Winter, C. Nastar, « Differential feature distribution maps for image segmentation and region queries in image databases », In *CBAIVL Workshop at CVPR'99, Fort Collins, Colorado, USA*, 1999.

[31] H. J. Wolfson, « Model-based object recognition », In *1st European Conference on Computer Vision*, p. 526-536, Lecture Notes in Computer Vision 427, Springer, 1990.

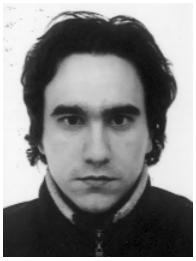
[32] H. J. Wolfson, « On curve matching », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, #5, 1990, p. 483-489.

[33] H. J. Wolfson, I. Rigoutsos, « Geometric hashing: an overview » , *IEEE Computational Science & Engineering*, October-December 1997, p. 10-21.

[34] C.T. Zahn, R.Z. Roskies. « Fourier descriptors for plane closed curves », *IEEE Transactions on Computers*, Vol. C-21, #3, 1972, p. 269-281.

LES AUTEURS

Pablo MUSÉ



Pablo Musé est né à Montevideo, Uruguay, en 1975. En 1999, il a obtenu le diplôme d'Ingénieur en Génie Électrique de l'Universidad de la Repûblica, Uruguay, et en 2001 le diplôme de DEA « Mathématiques, Vision et Apprentissage » de l'Ecole Normale Supérieure de Cachan. Actuellement il prépare une thèse en mathématiques appliquées au traitement d'images au CMLA, ENS-Cachan.

Frédéric SUR

Frédéric Sur est né en 1976. Ancien élève de l'École Normale Supérieure de Cachan, il prépare une thèse sur le traitement des images au CMLA.

Jean-Michel MOREL

Jean-Michel Morel est Professeur de Mathématiques à l'Ecole Normale Supérieure de Cachan. Il a obtenu son Doctorat de troisième cycle en 1980 et son doctorat d'état en 1985 de l'Université Pierre et Marie Curie. Depuis 1987, il travaille sur la modélisation mathématique et les méthodes numériques de l'analyse d'images digitales. Il est l'auteur avec S. Solimini d'un livre paru chez Birkhäuser (1994), Variational Methods in Image Segmentation. Il prépare actuellement avec F. Guichard un livre sur les E.D.P.s en analyse d'image, téléchargeable à
http://kyron.multimania.com/fg/
et, avec A. Desolneux et L. Moisan un livre sur les méthodes stochastiques en analyse d'images téléchargeable à
http://www.cmla.ens-cachan.fr/Utilisateurs/morel/PolyDEA2003.zip.