

Recherche dans les grandes bases de formes

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 : cet article présente une méthode d'appariement des formes contenues dans les images. Le codage des formes que nous considérons (les lignes de niveau significatives maximales) est local et invariant par similitude ou transformation affine. L'approche retenue permet une décision sans paramètre. Des résultats expérimentaux sur la recherche de logos sont présentés.

Abstract: this paper presents a matching algorithm for shapes in images. The shapes we consider are maximal meaningful level lines. Shape coding is local and invariant under similitude or affine transformation. Our decision rule is parameter-free. Experimental results for logo recognition are shown.

Introduction

Si citer l'explosion du multimédia et d'internet pour motiver les recherches en traitement d'images tient du lieu commun, il est indéniable que le support de l'information est désormais visuel. L'archivage est nécessaire : les immenses bases de photographies destinées aux médias n'ont une valeur commerciale qu'à cette condition. Deux problèmes successifs se posent donc. Quelles sont les caractéristiques pertinentes permettant d'indexer les images ? De quelle manière les comparer ?

Nous présentons d'abord les approches retenues par quelques équipes de recherche (section 1). Cette étude ne vise pas à l'exhaustivité : elle est essentiellement basée sur les travaux les plus significatifs disponibles sur le web. Néanmoins, elle met en valeur une notion cruciale : la notion de forme. En effet, le calcul des caractéristiques est basé soit sur des points d'intérêts (généralement des détecteurs de coins : ce sont bien des morceaux de formes), soit sur des zones homogènes de l'image. Le principal inconvénient de ces approches est la difficulté d'extraction. De plus, ces formes ne sont pas liées à la géométrie de l'image. Nous proposons d'utiliser les lignes de niveau significatives de l'image, qui sont naturellement des éléments géométriques. Un codage local de ces lignes permet leur indexation (section 2). Nous définissons également la notion d'appariement significatif, qui a la très intéressante propriété de ne pas nécessiter de paramètre (section 3). Des résultats expérimentaux sont également donnés.

1 État de la recherche

Voici un rapide état de l'art à partir des travaux disponibles sur le web.

1.1 Logiciels commerciaux

Dans la plupart des cas, l'information fournie par les entreprises sur leurs logiciels est insuffisante. Les algorithmes mis en œuvre sont décrits de manière très approximative.

On peut citer les logiciels *Excalibur Visual Retrieval-Ware* (Excalibur Corp.), *ImageFinder* (Attrasoft), *Imatch* (MW Labs), *QBIC* (Query by Image Content, IBM), *VIR Image Engine* (Virage), les logiciels développés par la société LookThatUp (nouvellement LTU Technologies)... Le problème généralement traité est celui de la recherche d'images similaires à une image donnée. Les données extraites des images sont des attributs de couleurs (histogrammes globaux et locaux), de textures (réponses à des filtres de Gabor), ou de formes (moments par exemple). La définition d'une distance entre ces attributs permet de définir une notion de «similarité» entre deux images.

Pour des comparaisons entre ces différents logiciels, voir [1].

Ces techniques sont analogues à celles développées par les équipes académiques dont nous exposons quelques travaux dans la partie suivante.

1.2 Recherche académique

Pour indexer des images il faut extraire l'information pertinente, et celle-ci ne doit pas être trop volumineuse afin de permettre une recherche rapide. Deux approches sont possibles : l'utilisation de points d'intérêt (section 1.2.1) ou l'utilisation de régions cohérentes (section 1.2.2). D'autre part, si l'on suppose le problème de l'extraction des formes présentes dans une image résolu, on peut s'intéresser à l'indexation directe des formes (section 1.2.3).

1.2.1 Caractéristiques de points d'intérêt

Movi (Modélisation pour la Vision, Inria Grenoble). Le problème est celui de la recherche d'images semblables à une image donnée. Les points d'intérêt sont les coins présents dans les images ; ils sont repérés à l'aide de détecteurs dédiés. Les caractéristiques sont des vecteurs d'invariants différentiels à différentes échelles d'observation, ce qui assure une pseudo-invariance par similitude. La recherche d'images similaires se fait sur la base d'un vote : les appariements entre caractéristiques sont comptés. Une contrainte de cohérence locale permet d'éviter certains faux appariements. L'algorithme de vote peut être remplacé par une approche probabiliste : une probabilité d'appariement (incorporant la cohérence spatiale) entre une image de la base et le modèle est définie. Cet algorithme a un bon comportement vis-à-vis des transformations géométriques (similitudes), et gère bien le problème des occlusions. Références : [2], [3].

Kiwi (Key-points Indexing Web Interface, Insa Lyon). Les caractéristiques utilisées ici prennent en compte des invariants différentiels, la géométrie de l'image, les réponses à des filtres de Gabor. . . Un histogramme sur l'ensemble des points d'intérêt est calculé pour chaque caractéristique. Les histogrammes de l'image-requête et ceux de la base sont ensuite comparés par corrélation. Enfin, les images de la base sont classées suivant une moyenne pondérée des corrélations. Une interaction est possible avec l'utilisateur par le choix des différents poids. Le point fort de cet algorithme réside dans le temps d'exécution. Références : [4], [5].

1.2.2 Caractéristiques de régions

Imedia (Images et Multimedia, Inria Rocquencourt). Le logiciel *Surfimage* est le précurseur de la solution proposée par la société LookThatUp. Il permet de déterminer des images semblables à une requête. L'indexation et la recherche se font selon différents types de signatures : signatures bas-niveau (histogrammes de couleurs et d'orientations, transformées de Fourier et en ondelettes. . .) et signatures haut-niveau («images-propres», «flexible images», ISS. . .). Les plus proches voisins d'une image requête sont recherchés dans la base. Une boucle de pertinence avec interaction de l'utilisateur permet de définir une «signature virtuelle» (censée représenter l'image-souhait de celui-ci) qui est comparée aux signatures de la base. Toujours dans le cadre de ce projet, une autre approche a été proposée : les images sont segmentées en régions cohérentes vis-à-vis des signatures précédentes. La recherche d'images semblables à une requête se fait alors sur la base de signatures globales associées à ces régions. Références : [6], [7].

Blobworld (UC Berkeley Digital Library Project)

En chaque pixel de chaque image est calculé un vecteur de caractéristiques contenant des composantes associées

à la couleur, des descripteurs de texture et la position. La distribution de ces caractéristiques est alors modélisée comme une superposition de gaussiennes, ce qui permet de définir des classes et fournit une segmentation de l'image. Les descripteurs de chaque région sont les histogrammes de couleurs et de textures. Référence : [8].

1.2.3 Recherche de formes

Squid (Shape Queries Using Image Databases, Centre for Vision, Speech and Signal Processing, University of Surrey, UK).

Cette équipe travaille dans le cadre de la théorie du scale-space, et s'intéresse à la mise en correspondance de formes (des courbes de Jordan). Les courbes (paramétrage canonique s) subissent un lissage gaussien (variance σ) et sont renormalisées (*curvature scale space*) ; l'ensemble des points en lesquels la courbure change de signe est représenté dans l'espace (σ, s) . Cette représentation est robuste par rapport aux changements d'échelle, au bruit (grâce au lissage) et aux transformations rigides. La mise en correspondance des courbes est réalisée par appariement entre ces représentations. Références : [9], [10].

Photobook (MIT Media Laboratory).

Les images sont représentées par les formes qu'elles contiennent, leur apparence (représentation en images-propres) et les textures, et la recherche utilise l'une de ces descriptions. L'approche utilisée pour décrire les formes est issue d'un modèle physique : les formes sont caractérisées par leurs modes de déformation, et l'énergie nécessaire pour déformer une forme en une autre quantifie la qualité des appariements. Références : [11], [12].

Recherche de logos. Les logos ne sont généralement pas texturés, et sont bien représentés par leurs contours : la recherche de logos entre bien dans le cadre de la recherche de formes.

- *Negative Shape Features*

Un algorithme est présenté dans [13] et [14]. Le but est de trouver des logos dans une base semblables à un logo donné. Les logos (des images binaires) sont segmentés par composantes connexes. Un vecteur de caractéristiques est extrait pour chacune des composantes. Ce vecteur est composé de quatre descripteurs globaux : moment d'inertie, «circularité», «rectangularité», «excentricité» ; et de trois descripteurs locaux : proportion de «trous» dans la région, de «sauts verticaux» et de «sauts horizontaux». La recherche d'un logo semblable à un logo-requête s'appuie sur une distance (au sens des caractéristiques) entre les régions des deux logos.

- *ARTISAN*

Une autre approche est celle d'*ARTISAN* (Automatic Retrieval of Trademark Images by Shape ANalysis, Université de Northumbria, Newcastle, UK). *ARTISAN* a été

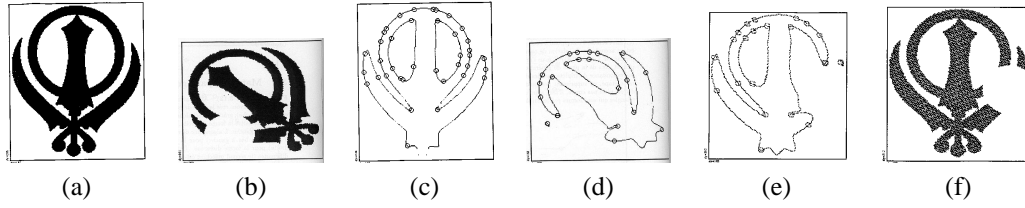


FIG. 1 Exemple de matching par geometric hashing. (a) : image originale, (b) : image distordue et bruitée, (c) : échelle 6 de (a), (d) : échelle 4 de (b), (e) : image de (d) par la transformation estimée, et (f) : image de (b) par cette transformation.

conçu spécialement pour l'office d'enregistrement de brevets britannique. Étant donné un nouveau logo, *ARTISAN* permet de trouver les logos les plus semblables selon certains critères. En suivant les principes de la Gestalt, on suppose que les éléments des images sont perçus comme des groupes, et on essaye de les représenter explicitement tels quels. L'algorithme implémenté dans *ARTISAN* est le suivant : approximation des formes du logo par des droites et des arcs de cercle, puis groupement en familles de formes proches, enfin extraction de caractéristiques globales des formes à 3 niveaux (image en entier, chaque famille de formes, chaque forme individuelle). Les caractéristiques utilisées sont la circularité, différents moments, etc. Référence : [15].

- scale space et geometric hashing

Dans sa thèse de doctorat, Thierry Cohignac s'intéresse au problème de la reconnaissance de formes planes, définies comme les composantes connexes d'images binaires. Son but est de trouver une représentation robuste affine-invariante des formes. Il propose deux méthodes d'appariement, l'une globale et l'autre semi-locale. Cette dernière permet de gérer le problème de l'occlusion. Présentons rapidement les grandes étapes de ces algorithmes (pour les détails, voir [16]).

Reconnaissance globale : les formes de la base sont normalisées (normalisation de matrices d'inertie), puis des paramètres globaux (moments) sont extraits. Des hypothèses d'appariement sont générées par comparaison des moments, enfin un recalage permet de vérifier les hypothèses.

Reconnaissance semi-locale : les formes subissent un lissage affine invariant. Puis, si F est la forme initiale et F_t est la forme obtenue après lissage à l'échelle t , on calcule les *surfaces caractéristiques*, définies comme l'ensemble des formes $\{F_{3t_0+i} - F_{t_0}\}_{i=1\dots N}$. Ensuite on extrait des points caractéristiques à ces différentes échelles : les extrema de courbure des surfaces caractéristiques. Avec ces points caractéristiques on construit une table de hachage multi-échelle. Le «geometric hashing» permet d'apparier les formes ; à l'aide d'un recalage on vérifie la fiabilité des appariements.

Les images de la figure 1 illustrent le résultat d'un appariement par *geometric hashing*. Le but est de reconnaître (b) comme la version distordue et avec occlusion de (a). L'algorithme trouve une correspondance entre l'échelle 6 pour la forme originale (1(c)), et l'échelle 4

pour la forme distordue (1(d)).

2 Comparaison d'images par leurs formes

Les algorithmes développés par José-Luis Lisani ([19]) représentent le point de départ de notre travail. Rappelons les quatre étapes de codage et de recherche d'une courbe.

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. Mais la plupart de ces formes n'a pas d'intérêt sémantique (figure 2(b)). Ainsi, l'énorme quantité de lignes obtenues oblige à faire une sélection des lignes perceptuellement «importantes». Pour sélectionner ces lignes nous utilisons la technique présentée par Desolneux et al. dans [17], qui définit les lignes significatives comme étant suffisamment contrastées et longues, au sens du principe de Helmholtz (figure 2(c)). On observe que souvent les objets représentés dans l'image par les lignes significatives présentent des bords épais, constitués par un ensemble de lignes de niveau 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 dans [21]) ces structures correspondent à des branches où chaque nœud n'a qu'un seul fils, et le niveau de gris des courbes de toute la branche est soit croissant soit décroissant avec la profondeur dans l'arbre. Ainsi, afin de représenter les bords avec une seule ligne, on choisit la ligne la plus significative au sens contraste-longueur. La figure 2(d) montre l'ensemble des lignes de niveau obtenues finalement, que Desolneux définit comme *lignes significatives maximales*. Ces lignes fournissent une représentation compacte des formes contenues dans l'image (notez la réduction dans le nombre des lignes).

Une fois les lignes extraites, nous appliquons à cet ensemble de lignes le schéma rapide de Moisan ([20]) pour le lissage invariant par transformation affine ([22]) :

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

où x est un point d'une ligne de niveau, $\text{Curv}(x)$ la cour-

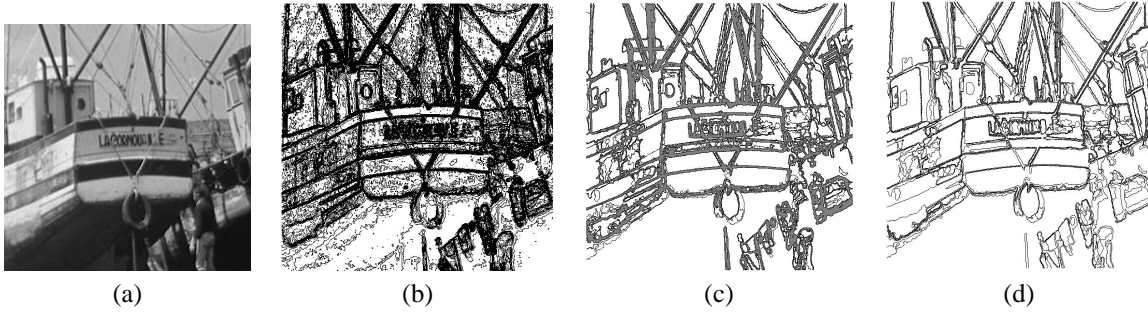


FIG. 2 Extraction des lignes de niveau significatives. (a) image originale, (b) lignes tous les 10 niveaux de gris (5479 lignes), (c) lignes de niveau significatives (4342 lignes), (d) lignes significatives maximales (296 lignes).

bure et $\vec{n}(x)$ la normale à la courbe, orientée vers la concavité.

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 (invariant euclidien ou affine), on définit des repères locaux pour l'ensemble des lignes de niveau issu de l'étape précédente (figure 3). Ceci permet de définir des codes pour des 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 $\{(x_i, y_i), 1 \leq i \leq N\}$.¹ Le caractère local du codage est une propriété fondamentale, notamment pour comparer des images présentant des occlusions. Ainsi, à chaque image on associe un dictionnaire qui contient tous les codes extraits de ses lignes de niveau significatives.

3. *Comparaison grossière des dictionnaires : l'étape de pré-appariement.* Chaque code du dictionnaire D_1 de l'image I_1 est comparé avec tous les codes du dictionnaire D_2 de l'image I_2 . Si $|x_i^1 - x_i^2| < \eta_n$ et $|y_i^1 - y_i^2| < \eta_n$ pour tout $i \in \{1, \dots, N\}$, nous considérons que les deux morceaux de courbe constituent un «pré-appariement», *i.e.* sont des candidats pour être appariés. Remarquons que η_n est une distance dans le repère normalisé; cette valeur ne représente aucune distance «physique» entre les points des courbes à mettre en correspondance. L'idée de cette étape étant de garder, pour chaque code de D_1 , l'ensemble des codes de D_2 qui sont *suffisamment* proches, ce seuil ne doit pas être trop exigeant. L'avantage de l'étape de pré-appariement est qu'elle permet de diminuer le temps de calcul, car l'étape d'appariement est bien plus lourde.

4. *Extension des pré-appariements.* Dans cette étape, pour tous les morceaux de courbe qui constituent un pré-appariement, nous calculons la transformation qui mène du i -ième morceau de courbe de I_1 (C_i^1) au j -ième morceau de courbe de I_2 (C_j^2). Ceci nous permet de mettre C_1 et C_2 dans le référentiel de I_2 , et de considérer maintenant la distance entre ces morceaux de courbe dans

l'image. Pour ce faire, trois paramètres sont en jeu : η_{I_2} , L_{min} et θ_{min} .

- η_{I_2} contrôle la distance en pixels entre les points correspondants des deux codes, dénormalisés dans l'image I_2 .
- L_{min} contrôle la longueur d'arc minimale exigée pour les deux morceaux à comparer (dans leur référentiel image d'origine). En effet, on étend les morceaux au delà de leurs limites initiales, jusqu'à ce que la distance entre deux points correspondants dépasse η_{I_2} . Si cette longueur est inférieure à L_{min} , nous ne prenons pas en compte l'appariement.
- Les morceaux de courbe correspondants dont la longueur d'arc dépasse L_{min} sont considérés comme possibles appariements si leurs variation d'angle dépasse θ_{min} .

Un exemple de comparaison d'images : recherche d'un logo dans une publicité. Nous nous proposons ici de chercher un logo dans une image publicitaire, en partant d'un logo prototype d'assez bonne qualité. La figure 4 montre les images utilisées pour cette expérience. Les lignes de niveau extraites sont présentées dans la figure 5.

La figure 6 montre les appariements retrouvés. Dans ce cas, aucun faux appariement ne s'est produit; l'image de droite montre en détail la zone de la publicité qui contient tous les appariements trouvés. Les paramètres utilisés ici sont $L_{min} = 35$ pixels, $\theta_{min} = 200$ degrés et $\eta_{I_2} = 2$ pixels.

La détermination des appariements entre lignes entières par vote du nombre d'appariements entre morceaux de courbe aboutit aux résultats présentés dans la figure 7. Les lignes de niveau des trous des lettres P et A n'ont pas été appariés, car leur longueur d'arc est inférieure à 35 pixels.

¹ Pour chaque code, nous stockons aussi en mémoire la transformation effectuée pour la normalisation. Cette information peut s'avérer utile ultérieurement, pour vérifier, par exemple, la cohérence de ces transformations.

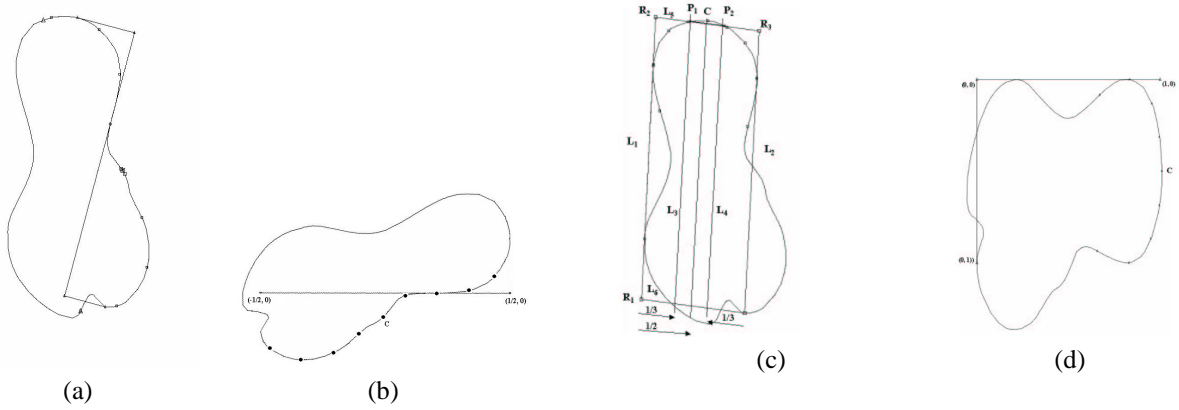


FIG. 3 (a) repère invariant par similitude sur la courbe originale, (b) repère et courbe normalisés, (c) repère affine invariant sur la courbe originale, (d) repère et courbe normalisés.



FIG. 4 À gauche, logo à chercher dans l'image de droite.

3 Introduction d'un seuil de rejet / acceptation

La principale faiblesse de la méthode que nous venons de décrire est qu'elle met en jeu trop de paramètres : la longueur des morceaux normalisés, la densité de l'échantillonnage, η_n , η_{I_2} , L_{min} et θ_{min} .

Dans certains types de problèmes, la méthode décrite fournit des résultats satisfaisants. C'est le cas du recalage de paires stéréo ou de *frames* d'un film, et, de manière plus générale, de tous les problèmes où les paires de courbes qui sont en correspondance sont quasiment les mêmes, à un certain degré d'invariance près. En effet, dans ce genre de problèmes, il suffit de choisir les paramètres assez restrictivement pour obtenir un très grand nombre de vrais appariements. Et même si parmi ces appariements il y en a des faux, leur proportion est faible et ils peuvent être rejetés en utilisant, par exemple, la cohérence des transformations entre courbes appariées.

Cependant, dans de nombreuses applications il peut arriver que les morceaux de courbe à mettre en correspondance soient assez différents, même à de forts degrés d'invariance près. Considérons par exemple le problème

de la recherche d'un logo donné dans une base d'images. Souvent, il existe plusieurs versions d'un même logo, que nous pouvons trouver semblables mais qui sont loin de correspondre à la même forme à une transformation affine près. Dans ce cas, si le choix des paramètres est assez restrictif, le seuillage peut être trop sévère et le logo ne sera pas détecté. D'autre part, si les seuils sont permissifs, la proportion de faux appariements risque de devenir trop élevée. Ainsi, la qualité du résultat dépend énormément du choix des paramètres, notamment du choix de η_{I_2} .

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

Le premier problème à résoudre est le choix de η_n , ou d'un autre paramètre équivalent qui tienne compte de la "distance" entre codes normalisés. Ce choix, au lieu d'être laissé à l'utilisateur, devrait être automatique ; il pourrait être déduit de la distribution des "distances" entre les codes de D_1 et D_2 . Le problème pourrait alors se formuler dans les termes suivants : étant donnés D_1 et D_2 , quels sont les appariements significatifs ? Nous

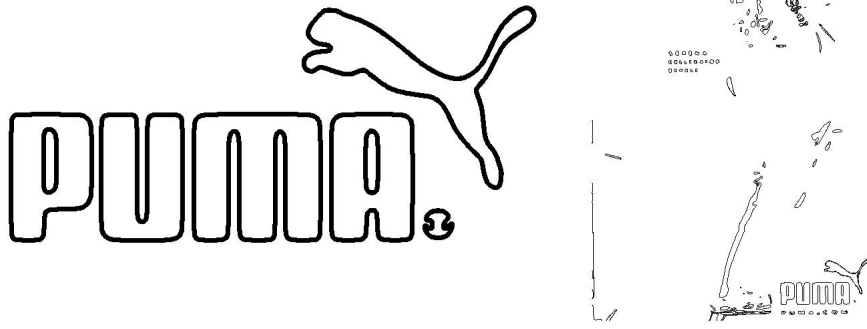


FIG. 5 Lignes de niveau maximales significatives lissées.

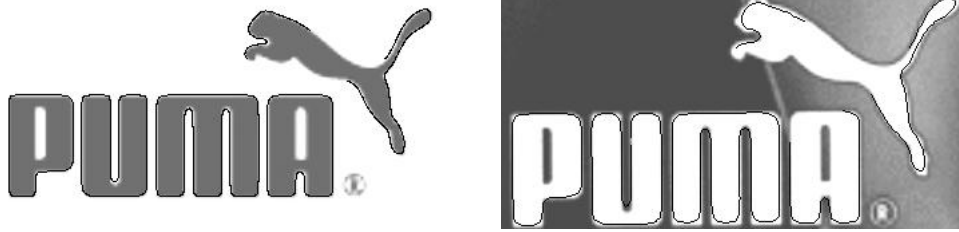


FIG. 6 Appariements étendus.

abordons ce problème dans la prochaine section.

3.1 Significativité des appariements de codes normalisés

Soit I_0 une image de requête et $\mathcal{B} = \{I_k, 1 \leq k \leq N_I\}$ une base d'images. À I_0 et à chaque I_k de la base correspond un dictionnaire de codes $D_k = \{C_j^k, 0 \leq j \leq N_k\}$.

Pour tout $k \in \{0, \dots, N_I\}$, les codes C_j^k sont composés d'un nombre fixe de points (x_i, y_i) :

$$C_j^k = \{(P_j^k)_i, 1 \leq i \leq N\}.$$

Notons $D_{\mathcal{B}} = \bigcup_{k=1}^{N_I} D_k$ l'ensemble de tous les

codes des images de la base \mathcal{B} , et $N_{\mathcal{B}}$ le nombre d'éléments dans $D_{\mathcal{B}}$.

Nous considérons les événements géométriques suivants :

- Évènement A : “un code de $D_{\mathcal{B}}$ est suffisamment proche d'un code donné de D_0 ”.
- Évènement B : “un code de $D_{\mathcal{B}}$ est suffisamment proche d'un code quelconque de D_0 ”.

La notion “suffisamment proche” est à prendre dans un contexte relatif : si tous les codes de $D_{\mathcal{B}}$ sont assez proches des codes de D_0 , nous nous intéressons à ceux qui sont les plus proches. Mais si tous les appariements sont mauvais, même les meilleurs seront mauvais, et si tous les appariements sont bons, en gardant les

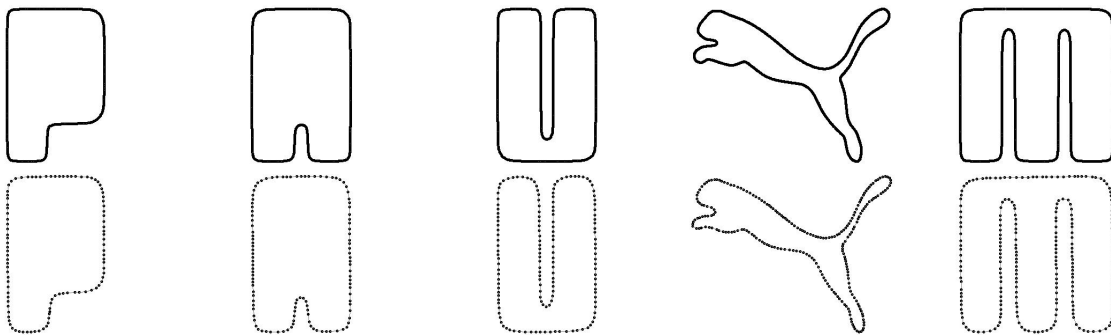


FIG. 7 Détermination des lignes de niveau qui s'apparient, par vote du nombre d'appariements de morceaux par courbe. La première ligne correspond aux courbes du logo, la deuxième ligne aux courbes de la publicité.

dix meilleurs on perdra beaucoup de bons appariements. L'idée est alors d'établir une mesure de proximité entre codes, dépendante du contexte, qui puisse répondre à ces exigences intuitives.

Avant de définir cette mesure de proximité, revenons un instant aux événements A et B . Considérer l'évènement A revient à chercher, pour chaque code C_j^0 , les codes C_j^B de D_B qui donnent les bons appariements. Dans le cas de l'évènement B , la recherche de bons appariements devient globale, et on est amené à trouver les bons appariements entre tous les codes de D_0 et tous les codes de D_B . Cette dernière option pose un problème qui peut être illustré par l'exemple suivant. Supposons que la base \mathcal{B} est composée d'une seule image I_1 et que cette image contient une forme qui se répète plusieurs fois dans l'image. Chacune de ces formes sera représentée par les mêmes codes, et s'il existe un code de D_0 très proche de ces codes, beaucoup de bons appariements associés aux autres codes de D_0 ne seront pas pris en compte. Cette remarque nous motive à considérer par la suite l'évènement A . Le code C_j^0 étant donné, nous le noterons C^0 .

3.1.1 Appariements significatifs

Définition 1 Deux codes normalisés de N points C^0 et C_j^B sont $\{\delta_1, \dots, \delta_N\}$ -proches si $\|(P^0)_i - (P_j^B)_i\|_2 \leq \delta_i \forall i \in \{1, \dots, N\}$.

Sous l'hypothèse d'indépendance des codes, la probabilité \mathcal{P} que C^0 et C_j^B soient $\{\delta_1, \dots, \delta_N\}$ -proches est donnée par la formule suivante :

$$\mathcal{P} = \prod_{i=1}^N H_i(\delta_i),$$

où $H_i(\delta_i)$ est la probabilité empirique (apprise sur la base d'images) que le i -ième point de C^0 et le i -ième point d'un code de D_B soient à une distance euclidienne plus petite que δ_i . Cette probabilité est donnée par C^0 et D_B :

$$H_i(\delta_i) = \frac{1}{N_B} \# \left\{ C_j^B / \|(P^0)_i - (P_j^B)_i\|_2 \leq \delta_i, 1 \leq j \leq N_B \right\}. \quad (1)$$

Pour pouvoir définir ce qu'est un appariement significatif, nous devons d'abord estimer le nombre de fois où C^0 et C_j^B sont $\{\delta_1, \dots, \delta_N\}$ -proches.

Définition 2 (Nombre de fausses alarmes) Soient $\delta_1, \dots, \delta_N$ fixés a priori. Le nombre de fausses alarmes de l'évènement " C^0 et C_j^B sont $\{\delta_1, \dots, \delta_N\}$ -proches" est défini par

$$NF(C^0, C_j^B) = N_B \mathcal{P} = N_B \prod_{i=1}^N H_i(\delta_i). \quad (2)$$

En suivant [18], nous définissons un appariement ε -significatif comme étant un appariement à nombre de fausses alarmes inférieur à ε .

Notation : nous notons $A(C^0, C_j^B)$ l'appariement entre les codes C^0 et C_j^B .

Définition 3 (Appariement ε -significatif) Supposons toujours $\delta_1, \dots, \delta_N$ fixés a priori. $A(C^0, C_j^B)$ est ε -significatif si :

$$NF(C^0, C_j^B) = N_B \prod_{i=1}^N H_i(\delta_i) \leq \varepsilon. \quad (3)$$

Nous allons maintenant donner une règle pour déterminer a priori les δ_i comme fonctions de ε . Pour les codes à N points, le nombre de fausses alarmes est fonction des N variables δ_i . Les seuils de significativité pour les distances δ_i (que nous noterons δ_i^*) ne sont pas alors uniques. Cependant, si nous considérons qu'il n'y a aucune raison pour que certains points d'un code soient plus significatifs que les autres, il est raisonnable d'ajouter la contrainte suivante :

$$N_B \times H_i(\delta_i) \leq \alpha \quad \forall i \in \{1, \dots, N\}, \quad (4)$$

α étant une constante à déterminer. En appliquant cette contrainte à l'équation 3 on obtient

$$N_B \prod_{i=1}^N H_i(\delta_i) \leq N_B \left(\frac{\alpha}{N_B} \right)^N \leq \varepsilon.$$

Ainsi, α doit nécessairement vérifier $(\alpha/N_B)^N \leq \varepsilon/N_B$. En prenant $\alpha/N_B = (\varepsilon/N_B)^{1/N}$, l'équation 4 devient :

$$H_i(\delta_i) \leq \left(\frac{\varepsilon}{N_B} \right)^{1/N} \quad \forall i \in \{1, \dots, N\}.$$

Les fonctions $\delta_i \mapsto H_i(\delta_i)$ étant croissantes, pour chaque $i \in \{1, \dots, N\}$ nous avons un unique δ_i^* défini par :

$$\delta_i^* = H_i^{(-1)} \left((\varepsilon/N_B)^{1/N} \right), \quad (5)$$

où $H_i^{(-1)}(\lambda) = \inf\{\delta_i, H_i(\delta_i) \geq \lambda\}$ est la fonction pseudo-inverse de H_i . Il en découle alors le résultat suivant :

Proposition 1 Sous l'hypothèse de symétrie

$$H_i(\delta_i) \leq \left(\frac{\varepsilon}{N_B} \right)^{1/N} \quad \forall i \in \{1, \dots, N\},$$

si C^0 et C_j^B sont $\{\delta_1^*, \dots, \delta_N^*\}$ -proches, alors $A(C^0, C_j^B)$ est ε -significatif.

3.1.2 Propriétés du nombre de fausses alarmes

Notons F_N la fonction :

$$F_N(\delta_1, \dots, \delta_N) = N_B \prod_{i=1}^N H_i(\delta_i), \quad (6)$$

où $\delta_1, \dots, \delta_N$ sont fixés a priori. La croissance des fonctions $\delta_i \mapsto H_i(\delta_i)$ et le fait que $H_i(\delta_i) \leq 1 \quad \forall \delta_i$ entraînent les propriétés suivantes :

- *Croissance en δ_i pour tout $i \in \{1, \dots, N\}$:*
si $\delta_i < \delta'_i$, alors $F_N(\delta_1, \dots, \delta_i, \dots, \delta_N) \leq F_N(\delta_1, \dots, \delta'_i, \dots, \delta_N) \quad \forall i \in \{1, \dots, N\}$.
- *Décroissance du nombre de fausses alarmes avec N :*
 $F_{N+1}(\delta_1, \dots, \delta_{N+1}) \leq F_N(\delta_1, \dots, \delta_N), \quad \forall \delta_{N+1} \geq 0$.

Ces propriétés sont particulièrement intéressantes : l'appariement est d'autant plus significatif que la précision (première propriété) et le nombre de points sur lequel il est basé (deuxième propriété) sont grands.

3.2 Appariements significatifs maximaux

Suivant toujours [18], définissons à présent la notion d'appariement significatif maximal.

Définition 4 (Appariement significatif maximal)

$A_p((C^0)_J, (C_j^B)_J)$ est maximal si :

1. les deux codes à J points ne contiennent pas des sous-codes qui s'apparient plus significativement :
 $\forall L \in \{N_{min}, \dots, J-1\}, NF_p((C^0)_L, (C_j^B)_L) \geq NF_p((C^0)_J, (C_j^B)_J)$,
2. les deux codes à J points ne sont pas des sous-codes de deux codes qui matchent plus significativement :
 $\forall L \in \{J+1, \dots, N_{max}\}, NF_p((C^0)_L, (C_j^B)_L) > NF_p((C^0)_J, (C_j^B)_J)$.

La proposition suivante assure qu'un appariement ε -significatif vérifiant la contrainte de symétrie est toujours maximal par rapport à tous ses appariements partiels.

Proposition 2 Soit $A_p((C^0)_{J-1}, (C_j^B)_{J-1})$ ε -significatif, vérifiant l'hypothèse de symétrie

$$H_i(\delta_i) \leq \left(\frac{\varepsilon}{N_B}\right)^{1/(J-1)} \quad \forall i \in \{1, \dots, J-1\}.$$

Alors, si le J -ième point vérifie

$$H_J(\delta_J) \leq \left(\frac{\varepsilon}{N_B}\right)^{1/J}, \quad (7)$$

$A_p((C^0)_J, (C_j^B)_J)$ est ε -significatif, vérifie l'hypothèse de symétrie, et

$$NF_p((C^0)_J, (C_j^B)_J) \leq NF_p((C^0)_{J-1}, (C_j^B)_{J-1}).$$

La preuve est évidente et repose sur les propriétés suivantes :

- $(\varepsilon/N_B)^{1/(J-1)} \leq (\varepsilon/N_B)^{1/J} \quad \forall J \geq 2$ (car $\varepsilon < N_B$).
- $H_i(\delta_i) \leq 1$.

Remarques

- Les conditions de proximité pour les points $\{(P^0)_1, \dots, (P^0)_J\}$ d'un code $(C^0)_J$ s'assouplissent quand le nombre de points augmente (i.e. quand on considère les codes $(C^0)_K, K > J$).
- Lorsqu'on étend un appariement pour trouver le significatif maximal, il suffit de l'étendre jusqu'à ce que le dernier point ne vérifie plus la condition (7). L'appariement significatif maximal sera celui associé à son plus long sous-code.

3.3 Application à la recherche de logos dans une base d'images : appariements significatifs

Dans toutes les expériences présentées dans cette section nous utilisons le codage invariant par similitude. Les morceaux de courbe codés sont ceux dont la longueur d'arc normalisée est 4. Les codes sont constitués de 19 points issus d'un échantillonnage régulier des morceaux de courbe normalisés. Pour la recherche d'appariements, nous avons fixé ε à 1.

Nous disposons de deux ensembles d'images. D'une part un ensemble de logos, qui seront les images de requête ; d'autre part une base d'images \mathcal{B} constituée de quatre publicités.

La première expérience consiste à chercher dans \mathcal{B} les codes à apparier avec les codes d'un logo donné. Dans \mathcal{B} nous incluons toujours la publicité du logo de requête. Dans la deuxième expérience le principe est le même, mais nous excluons de \mathcal{B} la publicité du logo de requête.

La figure 8 présente les logos et ses lignes significatives lissées. La figure 9 montre les cinq images dont quatre d'entre elles constitueront les bases \mathcal{B} .

expérience 1 : La publicité du logo est présente dans la base d'images.

Pour cette expérience la base \mathcal{B} est composée des images *pumapub*, *sonypub*, *supermanpub* et *poseidonpub*. Elle contient $N_B = 1351$ codes. Les résultats sont présentés dans les tableaux 1 et 2.

Les figures 10 et 11 montrent les deux appariements qui ne correspondent pas à la publicité du logo de requête. On observe que ces appariements sont en eux-mêmes raisonnables.

expérience 2 : La publicité du logo n'est pas présente dans la base d'images.

Dans cette expérience on utilise une base \mathcal{B} différente pour chaque logo de requête. De l'ensemble des cinq publicités *pumapub*, *sonypub*, *supermanpub*, *poseidonpub* et *evianpub* nous prenons les quatre qui ne représentent

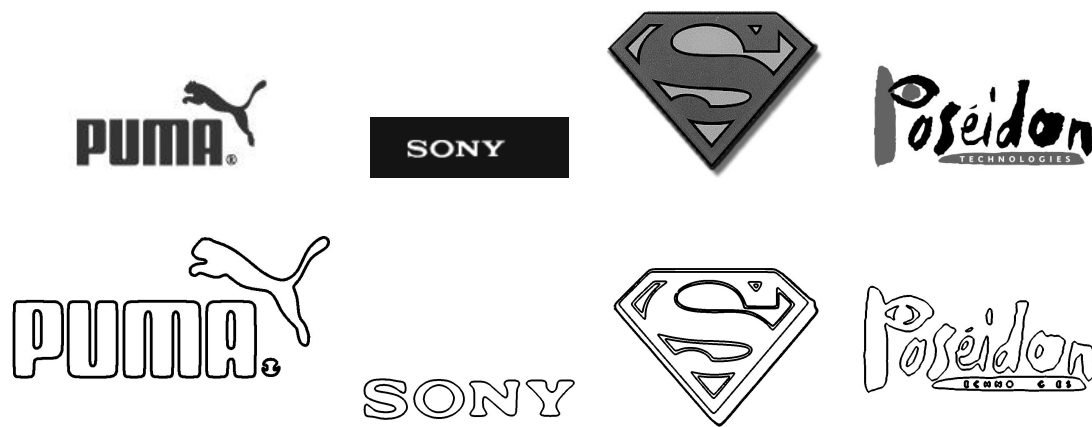


FIG. 8 Logos de requête : pumalogo, sonylogo, supermanlogo, poseidonlogo. Les images ne sont pas présentées avec leur taille originale ; le facteur de zoom appliqué n'est pas le même pour toutes les images.

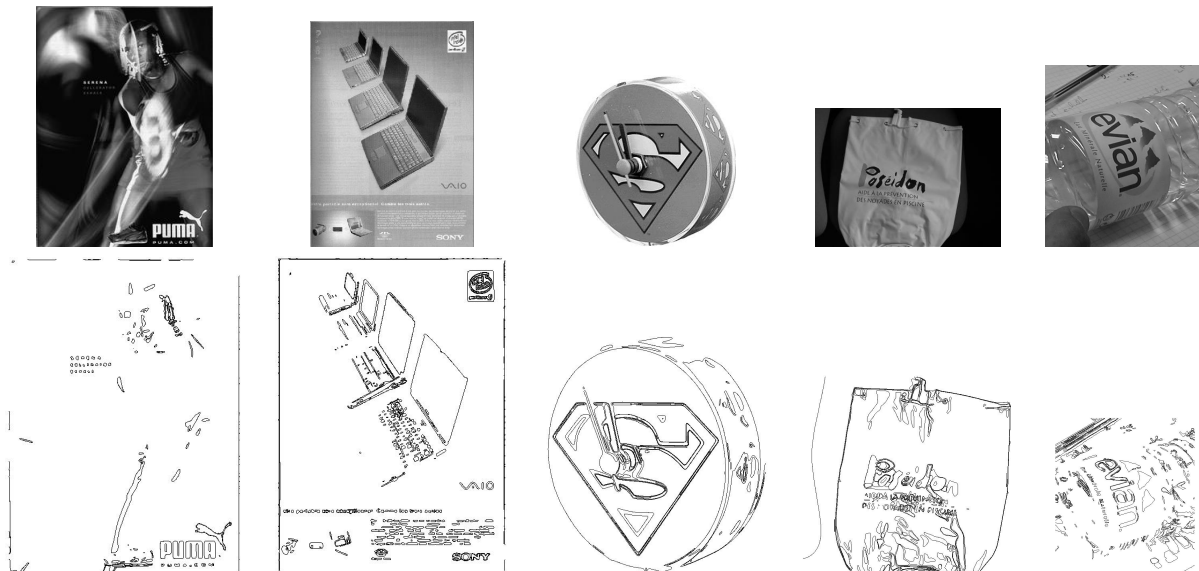


FIG. 9 Images de la base : pumapub, sonypub, supermanpub, evianpub. Leurs tailles ont été modifiées avec des zooms différents.

pas le logo. Les résultats sont présentés dans le tableau 3.

L'appariement accidentel obtenu pour *pumalogo* est le même que celui obtenu dans la première expérience, présenté dans la figure 10. De même pour l'appariement accidentel *sonylogo* (voir figure 11). Les figures 12 et 13 présentent les appariements accidentels pour *poseidonlogo*. On observe qu'en fait, ces appariements accidentels sont corrects du point de vue local (le caractère "N" correspond avec un "N", etc).

Conclusion

La recherche dans les bases de formes comme nous l'avons exposée est particulièrement prometteuse. Le problème est mieux posé que la recherche d'images indexées selon leurs points d'intérêts ou des segmentations sans réelle signification géométrique. De plus, l'introduction d'un seuil «universel» de rejet/acceptation des

appariements résout le problème de la décision. L'analyse statistique de l'espace des formes géométriques permettra d'améliorer encore nos résultats, tant au niveau de la qualité des appariements qu'au niveau des temps de calcul.

Références

- [1] C. VENTERS et M. COOPER, *A Review of Content-Based Image Retrieval Systems*, Manchester Visualization Center, University of Manchester, 1999.
- [2] C. SCHMID et R. MOHR, *Local greyvalue invariants for image retrieval*, Pattern Analysis and Machine Intelligence, 1997.
- [3] C. SCHMID, *A structured probabilistic model for recognition*, Computer Vision and Pattern Recognition, vol. 2, 485-490, June 1999.

logo	N_{logo}	$N_{matchings}$	N_{in}	N_{out}
pumalogo	28	33	32	1
sonylogo	29	23	22	1
supermanlogo	17	15	15	0
poseidonlogo	80	7	7	0

TAB. 1 Résultats de la première expérience. N_{logo} est le nombre de codes extraits du logo, $N_{matchings}$ est le nombre d'appariements significatifs trouvés, N_{in} est le nombre d'appariements trouvés dans la publicité du logo, N_{out} est le nombre d'appariements trouvés dans le reste des images.

logo	N_{logo}	$N_{codematch}$	N_{good}	N_{casual}
pumalogo	28	13	12	1
sonylogo	29	11	10	1
supermanlogo	17	3	3	0
poseidonlogo	80	6	6	0

TAB. 2 Résultats de la première expérience. N_{logo} est le nombre de codes extraits du logo, $N_{codematch}$ est le nombre de codes du logo qui ont au moins un appariement significatif parmi les codes de \mathcal{B} , N_{good} est le nombre de codes du logo qui trouvent des paires correctes, N_{casual} est le nombre de codes du logo qui sont appariés avec des codes qui ne le devraient pas.

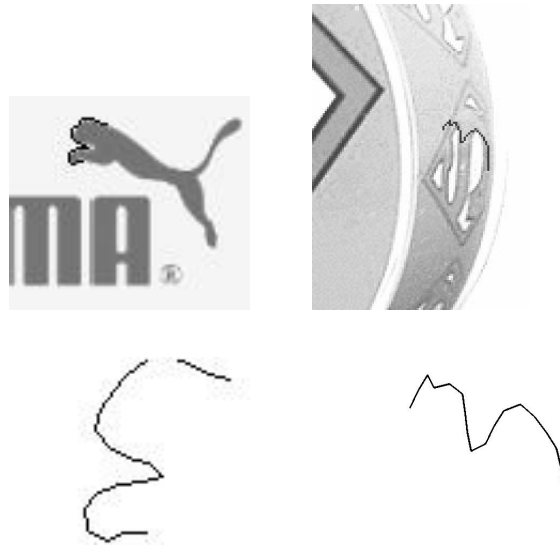


FIG. 10 Appariement accidentel pour l'image pumalogo.



FIG. 11 Appariement accidentel pour l'image sonylogo.

<i>logo</i>	N_{logo}	N_B	N_{casual}
pumalogo	28	1710	1
sonylogo	29	1535	1
supermanlogo	17	1435	0
poseidonlogo	80	1233	2

TAB. 3 Résultats de la deuxième expérience



FIG. 12 Premier appariement accidentel pour l'image poseidonlogo.

- [4] A. HEINRICHS, D. KOUBAROULIS, B. LEVIENAISE-OBADIA, P. ROVIDA et J.-M. JOLION, *Image indexing and content-based search using pre-attentive similarities*, RIAO2000, April 2000.
- [5] Ch. WOLF, J.-M. JOLION, W. KROPATSCH, et H. BISCHOF, *Content based image retrieval using interest points and texture features*, ICPR 2000, Barcelona, Spain, september 2000.
- [6] C. NASTAR, N. BOUEMAA, M. MITSCHKE, et C. MEILHAC, *Surfimage : un système flexible d'indexation et de recherche d'images*, Journées CNET, CORESA'98, Lannion, France, 9-10 juin 1998.
- [7] A. WINTER et C. NASTAR, *Differential feature distribution maps for image segmentation and region queries in image databases*, CBAIVL Workshop at CVPR'99, Fort Collins, Colorado, June 22nd, 1999.
- [8] C. CARSON, S. BELONGIE, H. GREENSPAN et J. MALIK, *Blobworld : Image segmentation using Expectation-Maximization and its application to image querying*, Third International Conference on Visual Information Systems, 1999.
- [9] F. MOKHTARIAN, S. ABBASI et J. KITTLER, *Efficient and robust retrieval by shape through curvature scale space*, proceedings of the first International Workshop on Image Databases and Multimedia Search, August 1996.
- [10] F. MOKHTARIAN, S. ABBASI et J. KITTLER, *Robust and efficient shape indexing through curvature scale space*, British Machine Vision Conference, september 1996.
- [11] A. PENTLAND, R.W. PICARD et S. SCLAROFF, *Photobook : Content-Based Manipulation of Image Databases*, Storage and Retrieval for Image and Video Databases II, Proc. SPIE, nb. 2185, 1994.
- [12] S. SCLAROFF et A. PENTLAND, *Modal Matching for Correspondence and Recognition*, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 17, No. 6, June 1995.
- [13] A. SOFFER et H. SAMET, *Negative shape features for image databases consisting of geographic symbols*, Third International Workshop on Visual Form, Capri, Italy, May, 1997.
- [14] A. SOFFER et H. SAMET, *Using negative shape features for logo similarity matching*, proceedings of the 14th International Conference on Pattern Recognition, volume I, Brisbane, Australia, August 1998.



FIG. 13 Deuxième appariement accidentel pour l'image poseidonlogo.

- [15] J.P. EAKINS, K. SHIELDS et J.M. BOARDMAN, *ARTISAN - a shape retrieval system based on boundary family indexing*, Storage and Retrieval Image and Video Databases IV, Proc. SPIE, pp. 17–28, 1996.
- [16] T. COHIGNAC, *Reconnaissance de formes planes*, thèse de doctorat, Ceremade, Université Paris IX Dauphine, 1994.
- [17] A. DESOLNEUX, L. MOISAN et J.M. MOREL, *Edge detection by Helmholtz principle*, Journal of Mathematical Imaging and Vision, vol. 14, pp. 271–284, 2001.
- [18] A. DESOLNEUX, L. MOISAN et J.M. MOREL, *Maximal Meaningful Events and Applications to Image Analysis*, preprint du CMLA no. 2000-22, disponible sur le site <http://www.cmla.ens-cachan.fr/Cmla/Publications/2000>.
- [19] J.L. LISANI, *Comparaison automatique d'images par leurs formes*, thèse de doctorat, Université Paris IX Dauphine, France, 2001.
- [20] L. MOISAN, *Affine plane curve evolution : a fully consistent scheme*, IEEE Trans. on Image Processing, 7 (3), pp. 411–420, 1998.
- [21] P. MONASSE, *Représentation morphologique d'images numériques et application au recalage*, thèse de doctorat, Université Paris IX Dauphine, France, 2000.
- [22] G. SAPIRO and A. TANNENBAUM, *Affine invariant scale-space*, International Journal of Computer Vision, 11 (1), pp. 25–44, August 1993.