

Approche pour la description, l'indexation et la recherche dans les grandes bases de formes

Hassan Silkan

Laboratoire LISQ Faculté des Sciences Dhar El Mahraz BP 1796 Atlas Fès Maroc

silkan_h@hotmail.com

Said Ouatik El Alaoui

Laboratoire LISQ Faculté des Sciences Dhar El Mahraz BP 1796 Atlas Fès Maroc

s_ouatik@yahoo.com

Abdelmounaime Lachkar

E.S.T.M Université Moulay Ismail Mekhnès Maroc

lachkar@estm.ac.ma

Abdelilah Elkharraz

Ecole Nationale de Commerce et de Gestion B.P 125.5 Tanger Maroc

kharraz_2003@yahoo.fr

Résumé

Dans ce travail, nous proposons une nouvelle approche pour la description, l'indexation et la recherche dans les grandes bases de formes. Nous apportons deux contributions. La première suggère un descripteur de forme invariant par rapport à la translation, l'homothétie et la rotation. Ce dernier est une version multi-échelle du descripteur proposé par Berretti et al. dont le principal inconvénient est la non invariance par rapport à la rotation. La seconde contribution consiste à proposer de stocker l'ensemble des index de toutes les formes de la base dans une seule structure d'index appelée M-tree. Cette structure en arbre est liée à toutes les formes de la base et non pas à chaque objet de celle-ci. Les résultats obtenus par l'application de notre approche sur une grande base de formes montrent l'intérêt de notre propos.

Abstract

In this work, we propose an approach for description, indexing and retrieval in large shapes databases. We introduce two contributions: the first one consists to propose a new shape descriptor which is invariant to translation, scale and rotation. This latter, is a multi-scale version of the Berretti et al. shape descriptor, which has the main drawback to not satisfy the rotation invariant criterion. In the second contribution, we propose to organize the indexes set of all shapes in the database using single index structure named M-tree, this tree index structure will be associated to all the shapes in the database instead to each one. By using a large shapes database, the obtained results highlight the effectiveness of our approach.

Mots-clés

descripteur invariant, indexation, M-tree, recherche par similarité, bases de formes

Keywords

invariant descriptor, indexing, M-tree, similarity search, shapes databases

1. INTRODUCTION

La quantité d'images produites et archivées chaque jour ne cesse de croître. Pour pouvoir exploiter ces archives colossales ainsi créées, des mécanismes pour indexer et pour retrouver les images stockées sont indispensables. Ceci nécessite le choix, d'une part, d'une représentation pertinente des images de la base par le biais de primitives visuelles significatives et fiables capables de décrire le contenu de la base et, d'autre part, d'une structure d'index efficace pour optimiser le temps de la recherche.

Généralement, les critères employés pour décrire une image sont des descripteurs de bas-niveau, appelés aussi vecteurs caractéristiques, telles que la couleur (Binaghi et Gagliardi et al., 1994), (Faloutsos, Flickner et al., 1993) et la texture (Liu et Picard, 1996), (Tamura, Mori et al., 1976). La forme (Del Bimbo et Pala, 1997), (Mokhtarian, Abbasi et al., 1996) occupe une place particulière parmi ces descripteurs dans la mesure où elle apporte une information moins ambiguë sur l'objet, car elle est plus proche de sa signification.

Différents descripteurs de formes existent dans la littérature (Veltkamp et Tanase, 2000). En 2000, Berretti et al. (Berretti, Del Bimbo et al., 2000) ont proposé un descripteur fondé sur les formes. Les formes sont partitionnées en des portions appelées « tokens ». Chaque token est représenté par deux attributs : la courbure et l'orientation. Comme il est mentionné par Zhang (Zhang et Lu, 2004), ce descripteur est invariant par rapport à la translation et l'homothétie, mais il présente l'inconvénient de ne pas être invariant par rapport à la rotation. Pour pallier ce problème, nous proposons une version multi-échelle du descripteur de Berretti et al. (Berretti, Del Bimbo et al., 2000) ayant l'avantage d'être invariante par rapport à la rotation. Il s'agit de :

- réaliser une rotation de tous les vecteurs d'orientation des tokens suivant un angle calculé en utilisant le principe d'équilibre des vecteurs de forces,
- calculer les deux attributs, courbure et orientation, à plusieurs valeurs d'échelles. Il s'agit alors d'une version multi-échelle du descripteur de Berretti et al. (Berretti, Del Bimbo et al., 2000) ayant l'avantage d'être invariante par rapport à la rotation.

Après l'étape de l'extraction des descripteurs la question qui se pose est de déterminer comment retrouver les images les plus similaires à une image requête. Plusieurs travaux ont été proposés. Ils sont fondés principalement sur les deux étapes suivantes :

- organiser les vecteurs caractéristiques dans des structures de données adaptées
- trouver les algorithmes de parcours efficaces pour améliorer le temps de la recherche.

Nous citons les méthodes d'accès spatial SAM (Spatial Access Methods) comme R-Tree (Guttman, 1984) et ses variantes (Beckmann, Kriegel et al., 1990), (Sellis, Roussopoulos et al., 1987) qui reposent sur l'utilisation d'une fonction de distance pour mesurer la similarité entre les vecteurs caractéristiques. Berretti et al. (Berretti, Del Bimbo et al., 2000) ont utilisé l'arbre métrique M-tree (Ciaccia, Patella et al., 1997). A chaque forme est associé un arbre. Deux mesures ont été définies, une pour calculer la similarité entre les tokens et une autre non métrique pour les formes. D'autres méthodes d'indexation ont été proposées. Mahmoudi et Daoudi (Mahmoudi et Daoudi, 2002) proposent de caractériser les objets 3D par des vues caractéristiques. Ils ont utilisé des index calculés à partir des vues et fondés sur le CSS (Curvature Scale Space). Ces index sont organisés autour d'une structure d'arbre M-tree. De même, ils ont associé un arbre M-tree à chaque vue de la base.

Pour une base de N formes, les deux derniers travaux cités précédemment produisent N arbres M-tree. Ainsi, la recherche est rapide au niveau de chaque forme par contre, elle est séquentielle sur la base. Pour éviter cet inconvénient, nous proposons d'utiliser l'arbre M-tree pour la totalité de la base et non pas pour chaque forme.

Le reste de cet article est organisé de la façon suivante. La section 2 présente le descripteur de forme proposé. Dans la section 3, nous décrivons la technique d'indexation adoptée. Les résultats

expérimentaux sont présentés en section 4. Dans la section 5, nous donnons une conclusion et quelques perspectives de ce travail.

2. Descripteur proposé

Nous proposons un descripteur invariant par rapport à la rotation. Ce dernier est une version multi-échelle du descripteur proposé par Berretti et al. (Berretti, Del Bimbo et al., 2000). Chaque token est représenté par deux attributs : la courbure et l'orientation.

Soit $f(u) = ((x(u), y(u)|u \in [0, T])$ la représentation paramétrique de la courbe correspondant à la forme, où u est l'abscisse curviligne et T est la longueur de la courbe. La famille des courbes $\{f(u, \sigma) / \sigma \geq 0\}$ obtenues par la convolution de $f(u)$ avec la gaussienne $g(u, \sigma)$ définie par :

$$g(u, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{u^2}{2\sigma^2}}$$

engendre l'espace d'échelle euclidienne. Soit aussi $K(u, \sigma)$ la courbure multi-échelle correspondant à la famille des courbes $\{f(u, \sigma) / \sigma \geq 0\}$:

$$K(u, \sigma) = \frac{x_t(u, \sigma)y_{tt}(u, \sigma) - x_{tt}(u, \sigma)y_t(u, \sigma)}{(x_t^2(u, \sigma) + y_t^2(u, \sigma))^{3/2}}$$

Tel que x_t, y_t et x_{tt}, y_{tt} sont respectivement les dérivées premières et secondes de x et y . Posons $P = \{P_{i(\sigma)}\}_{i=1}^N$ l'ensemble de points tels que $K(u, \sigma) = 0$. Si $K(u, \sigma)$ est continue entre deux points consécutifs $P_{i(\sigma)}$ et $P_{(i+1)(\sigma)}$, il existe toujours entre eux un extremum de $K(u, \sigma)$ nommé $m_{i(\sigma)}$ au point $P_{mi(\sigma)}$. Pour chaque valeur d'échelle σ on obtient une forme lissée qu'on partitionne en un ensemble de tokens, selon les points P_i . Chaque token i de $f(u, \sigma)$ est représenté par le vecteur $E_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)})$ (figure-1), où $m_{i(\sigma)}$ est la courbure de $P_{mi(\sigma)}$ comprise entre -180 et 180, et $O_{i(\sigma)}$ l'orientation du token en coordonnées polaires variant entre 0 et 360.

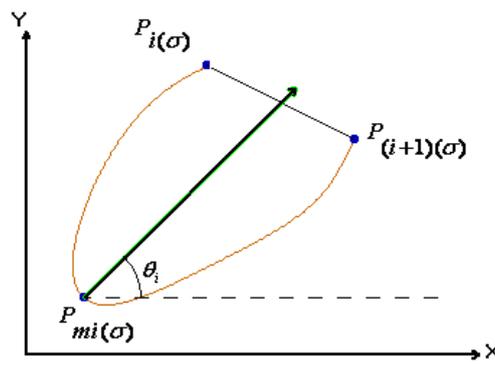


Figure 1. Représentation d'un token : l'angle θ_i mesure l'orientation du token.

Pour chaque valeur de lissage σ d'une forme, on obtient l'ensemble des vecteurs :

$$\Gamma_\sigma = \left\{ \vec{E}_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)}) \mid 1 \leq i \leq n_\sigma \right\}$$

Où n_σ est le nombre de tokens. Finalement le vecteur caractéristique obtenu pour une forme de la base est donné par :

$$\Gamma = \bigcup_{j=1}^M \Gamma_{\sigma_j} = \bigcup_{j=1}^M \left\langle \vec{E}_{i(\sigma_j)}(m_{i(\sigma_j)}, O_{i(\sigma_j)}) \mid 1 \leq i \leq n_{\sigma_j} \right\rangle$$

$$= \left\langle \vec{E}_i(m_i, O_i) \mid 1 \leq i \leq n \right\rangle$$

Où M est le nombre de valeurs de lissage, et $n = \sum_{j=1}^M n_{\sigma_j}$ est le nombre total des vecteurs. Notons que la courbure et l'orientation d'un token sont invariantes par la translation et par l'homothétie. Par conséquent, notre descripteur sera également invariant par rapport à ces transformations. Pour illustrer l'invariance par rapport à la rotation, prenons un exemple de deux orientations différentes d'un même token (cf. figure 2).

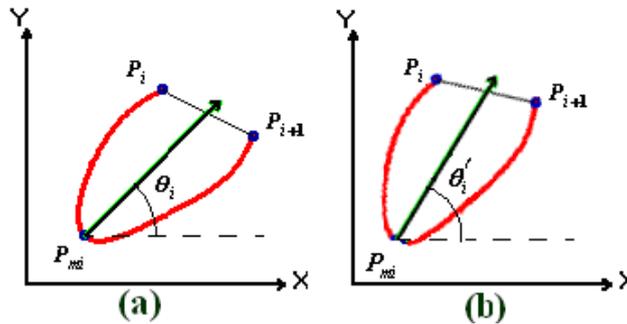


Figure 2. Un token représenté dans deux orientations différentes.

Soient $E1(m_i, \theta_i)$ et $E1(m'_i, \theta'_i)$ les descripteurs de token avant et après rotation (figure-2), $m_i = m'_i$ et $\theta_i \neq \theta'_i$, donc le descripteur considéré n'est pas invariant par rapport à la rotation.

Pour résoudre ce problème nous proposons la solution suivante : Soit \vec{a}_0 et \vec{b}_0 deux vecteurs non nuls, d'après le principe d'équilibre des vecteurs de forces, il existe un vecteur \vec{c}_0 tel que $\vec{a}_0 + \vec{b}_0 + \vec{c}_0 = \vec{0}$. Le vecteur \vec{c}_0 qui vérifie l'équilibre est appelé vecteur principal et sa direction par rapport à l'axe (OX) est appelée direction principale. Supposons que \vec{a}_θ et \vec{b}_θ sont les deux vecteurs obtenus après la rotation de \vec{a}_0 et \vec{b}_0 par un angle θ . Il existe un vecteur principal \vec{c}_θ qui vérifie $\vec{a}_\theta + \vec{b}_\theta + \vec{c}_\theta = \vec{0}$. La direction principale θ de \vec{c}_θ est déterminée par l'équation :

$$\theta = \arccos\left(\frac{\vec{c}_0 \cdot \vec{c}_\theta}{|\vec{c}_0| \cdot |\vec{c}_\theta|}\right)$$

Ce principe peut être généralisé à plusieurs vecteurs. En effet, soit $R(\theta)$ une transformation de rotation par un angle θ d'un vecteur donné. Pour un ensemble de vecteurs $\Lambda_0 = \{\vec{f}_i \mid 1 \leq i \leq N\}$, où N est le nombre de vecteurs et $\Lambda_\theta = \{\vec{g}_i \mid \vec{g}_i = R(\theta) \cdot \vec{f}_i, 1 \leq i \leq N\}$ sa rotation par un angle θ . Alors, si $\sum_{i=1}^N \vec{f}_i \neq \vec{0}$, il existe un vecteur principal \vec{F}_0 vérifiant $\sum_{i=1}^N \vec{f}_i + \vec{F}_0 = \vec{0}$. De même, si $\sum_{i=1}^N \vec{g}_i \neq \vec{0}$ il existe un vecteur principal \vec{F}_θ vérifiant l'équation (1)

$$\sum_{i=1}^N \vec{g}_i + \vec{F}_\theta = \vec{0} \quad (1)$$

La direction principale θ de \vec{F}_θ est donc déterminée par l'équation (2) :

$$\theta = \arccos\left(\frac{\vec{F}_0 \cdot \vec{F}_\theta}{|\vec{F}_0| \cdot |\vec{F}_\theta|}\right) \quad (2)$$

A partir de l'équation donnée en (1), on déduit que :

$$R(-\theta) \cdot \left[\sum_{i=1}^N \vec{g}_i + \vec{F}_\theta \right] = \sum_{i=1}^N [R(-\theta) \cdot \vec{g}_i] + R(-\theta) \cdot \vec{F}_\theta = \vec{0}$$

$$\text{Or,} \quad R(-\theta) \cdot \vec{F}_\theta = \vec{F}_0 \quad \text{et} \quad \vec{F}_0 = -\sum_{i=1}^N \vec{f}_i$$

$$\text{Donc} \quad \sum_{i=1}^N R(-\theta) \cdot \vec{g}_i = \sum_{i=1}^N \vec{f}_i \quad (3)$$

D'après la formule (3), on déduit que la direction initiale de l'ensemble $\Lambda_0 = \{\vec{f}_i | 1 \leq i \leq N\}$ peut être

déduite à partir de la rotation des vecteurs $\sum_{i=1}^N \vec{g}_i$ et \vec{F}_θ par l'angle $-\theta$. Cela peut être réalisé par l'application des deux formules suivantes :

$$R(-\theta) \cdot \vec{\Lambda}_\theta = \vec{\Lambda}_0 \quad (4)$$

$$R(-\theta) \cdot \vec{F}_\theta = \vec{F}_0 \quad (5)$$

Pour chaque forme de la base, nous proposons de calculer sa direction principale θ , ensuite réaliser une rotation de tous les vecteurs d'orientations des tokens par $-\theta$ de telle sorte que le vecteur principal des vecteurs caractéristiques de chaque forme de la base coïncide avec l'axe (OX). Ceci va permettre de rendre notre descripteur invariant par rapport à la rotation. En effet, pour deux descripteurs $\Lambda_{\theta 1}$ et $\Lambda_{\theta 2}$ d'une même forme avec deux orientations différentes, on calcule d'abord leurs vecteurs principaux respectivement $F_{\theta 1}$ et $F_{\theta 2}$. Ensuite, on détermine leurs directions principales θ_1 et θ_2 . Puis, on procède à une rotation de l'ensemble des vecteurs $\Lambda_{\theta 1}$ et $\Lambda_{\theta 2}$ respectivement par $-\theta_1$ et $-\theta_2$ en appliquant (4) et (5). On obtient alors pour $\Lambda_{\theta 1}$ et $\Lambda_{\theta 2}$ le même vecteur caractéristique Λ_0 ayant comme vecteur principal F_0 , i.e celui qui vérifie l'équation de l'équilibre et qui coïncide toujours avec l'axe (OX). La figure 3 présente le résultat d'un exemple produit par notre programme appliqué à une forme dans quatre orientations différentes. On remarque que pour les quatre orientations de la forme, on obtient le même descripteur (2d).

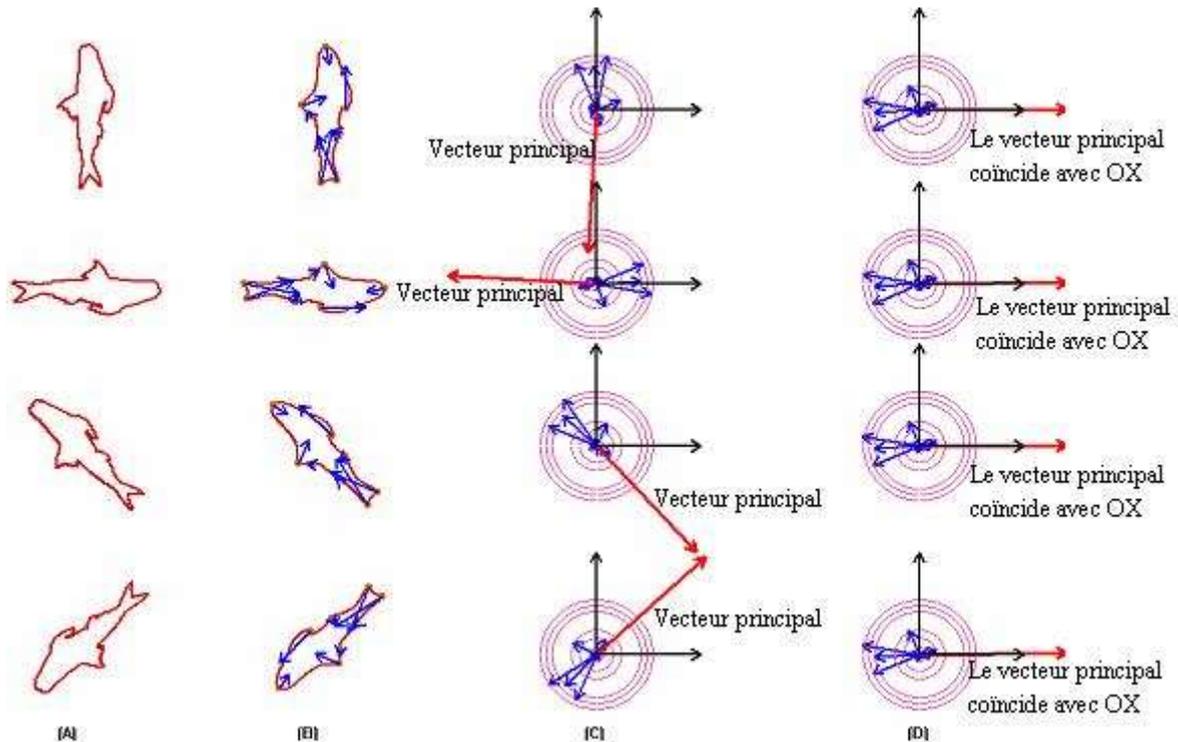


Figure 3. Invariance par rapport à la rotation : (A) une même forme dans quatre orientations différentes. (B) forme avec les vecteurs caractéristiques associés . (C) présentation des vecteurs caractéristiques et le vecteur principal dans un repère euclidien. (D) présentation des différents vecteurs après l'application d'une rotation $R(-\theta)$ à tous les vecteurs où θ est la direction principale.

L'algorithme utilisé pour calculer le descripteur est décrit ci-dessous.

Procédure calcul_descripteur(Forme F)

{ Soit A l'ensemble des vecteurs (courbure, orientation)

Soit A_σ l'ensemble des vecteurs caractéristiques

$A_\sigma \leftarrow \emptyset ; A \leftarrow \emptyset ;$

Pour chaque valeur de lissage σ faire

Pour chaque token f_i de F

{

Calculer $E_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)})$

Ajouter $E_{i(\sigma)}$ à A.

}

Calculer la direction principale θ de l'ensemble A.

Pour chaque vecteur E_i de A faire

{

$E'_i \leftarrow R(-\theta).E_i$

Ajouter E'_i à A_σ

}

}

3. Méthode d'indexation

Nous avons choisi une méthode d'indexation fondée sur les arbres métriques (Ulmanm, 1991). Elle nécessite la définition d'un espace métrique $M=(D,d)$ pour mesurer la similarité entre les formes où D est le domaine de vecteurs caractéristiques et d la fonction de distance vérifiant les trois axiomes suivantes :

- $d(O_x, O_y) = d(O_y, O_x)$ (symétrie)
- $d(O_x, O_y) > 0$ si $O_x \neq O_y$ et $d(O_x, O_x) = 0$ (Positivité)
- $d(O_x, O_y) \leq d(O_x, O_z) + d(O_z, O_y)$ (inégalité triangulaire)

3.1. Présentation de la structure d'index

L'arbre M-tree organise l'espace de vecteurs caractéristiques en un ensemble de régions (ou clusters) imbriquées. Chacune correspond à un nœud de l'arbre et regroupe un ensemble de formes similaires (ou très rapprochées). Il est composé de nœuds internes et de nœuds feuilles. Chaque nœud comporte un ensemble de M entrées au maximum (la capacité du nœud).

Toute entrée d'un nœud correspond à une forme de la base. Sa structure dépend de la nature de nœud.

L'entrée d'une feuille est définie par $\text{entry}(F_j) = [F_j, \text{oid}(F_j), d(F_j; P(F_j))]$ où F_j sont les descripteurs de la forme F_j ; $\text{oid}(F_j)$ est l'identifiant de la forme (nom du fichier image) et $d(F_j; P(F_j))$ désigne la distance entre F_j et l'objet père $P(F_j)$.

L'entrée d'un nœud interne est définie par $\text{entry}(F_r) = [F_r, \text{ptr}(T(F_r)), r(F_r), d(F_r; P(F_r))]$ où F_r sont les descripteurs de la forme F_r (routing object), $\text{ptr}(T(F_r))$ est le pointeur sur un sous arbre $T(F_r)$, $r(F_r)$ désigne le rayon de couverture, c'est la distance maximale entre F_r et les formes de la sous région et, $d(F_r; P(F_r))$ indique la distance entre F_r et l'objet père $P(F_r)$.

L'arbre est construit par insertions successives. Insérer un nouvel objet (une forme) consiste à trouver le nœud feuille le plus convenable pour ajouter cet objet. Ceci entraîne éventuellement la décomposition du nœud feuille s'il est saturé (invocation de la fonction Split). Le découpage nécessite deux méthodes « Promote » et « Partition ». La méthode Promote consiste à choisir deux objets internes O_{p1} et O_{p2} à insérer dans le nœud père N_p , tandis que la méthode Partition consiste à partitionner les $M+1$ entées en deux sous-ensembles disjoints.

3.2. Recherche par similarité dans M-tree

Pour réduire au minimum le nombre de nœuds consultés et le nombre des distances calculées, toutes les informations concernant ces distances sont stockées dans les nœuds de l'arbre. Dans ce travail, deux types de requêtes de similarité sont développés : requête par intervalle (range query) et k-plus proches voisins (k-nearest neighbors query).

Soient $M = (D,d)$ un espace métrique, et Q un objet requête appartient à D .

- Requête intervalle : Soit $r(Q)$ la distance maximale de recherche. $\text{Range}(Q,r(Q))$ est une requête qui sélectionne tous les objets indexés O_j tels que $d(O_j, Q) \leq r(Q)$.
- k plus proches voisins : Soit k un nombre entier tel que $k \geq 1$, Query-NN(Q,k) sélectionne les k objets indexés ayant la plus petite distance de Q .

Comme le nombre de vecteurs caractéristiques d'une forme requête et celui d'une forme cible ne sont pas toujours égaux, nous avons utilisé la distance de Hausdorff définie entre les ensembles (Remolar, C hover et al., 2002) pour mesurer la similarité entre deux formes.

Soient $X = \{\vec{x}_i(m_i, O_{i1}) / 1 \leq i \leq n\}$ l'ensemble de descripteurs d'une forme F et $Y = \{\vec{y}_j(m_j, O_{j1}) / 1 \leq j \leq m\}$ l'ensemble de descripteurs d'une forme requête Q .

La distance de Hausdorff entre X et Y est définie par :

$$d(X, Y) = \frac{\sum_{i=1}^n d(x_i, Y) + \sum_{j=1}^m d(y_j, X)}{|X| + |Y|}$$

Où:

$|X|$ est le cardinal de X .

$d(x_i, Y) = \min_{y_j \in Y} d(x_i, y_j)$ est la distance entre un vecteur x_i de X et l'ensemble Y tel que $d(x_i, y_j)$

$$d(x_i, y_j) = \alpha \cdot |m_{x_i} - m_{y_j}| + (1 - \alpha) \cdot |O_{x_i} - O_{y_j}|$$

est la distance entre deux vecteurs donnée par :

Afin de comparer deux vecteurs x_i et y_j tout en donnant la même importance pour chaque attribut (courbure et orientation), nous prenons $\alpha = 1/2$.

4. Résultats

Pour illustrer l'intérêt de notre approche, nous avons développé une application en Visual C++ permettant à l'utilisateur d'interroger une grande base de formes à travers une interface graphique conviviale.

4.1. Génération de la base de formes

Pour pouvoir tester et valider notre système d'indexation et de recherche, nous avons pris la base de formes utilisée dans <http://www.ee.surrey.ac.uk/Research/VSSP/imagenet> et nous l'avons enrichie en générant d'autres formes. Ainsi, nous avons produit pour chaque forme de la base, 10 formes avec différentes orientations (cf. figure 4). L'objectif est double. Nous souhaitons, d'une part, valider notre approche sur une base de taille très importante, et d'autre part, prouver expérimentalement l'invariance de notre descripteur par rapport à la rotation.

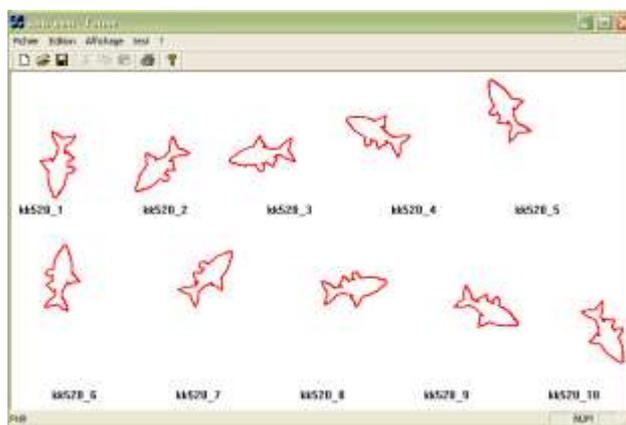


Figure 4. Exemple de génération de 10 formes ayant des orientations différentes.

4.2. Construction de l'arbre

La construction de l'arbre est effectuée d'une manière dynamique de bas vers le haut. Chaque entrée d'un nœud de l'arbre stocke un ensemble de vecteurs caractéristiques correspondant à une forme de la base. La figure 6 montre un exemple de construction de l'arbre pour 10 formes de la figure 5 suivante :

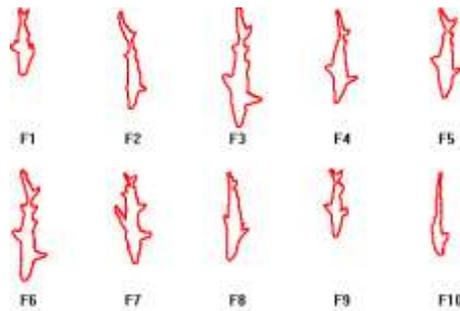


Figure 5.– Exemples de formes de la base

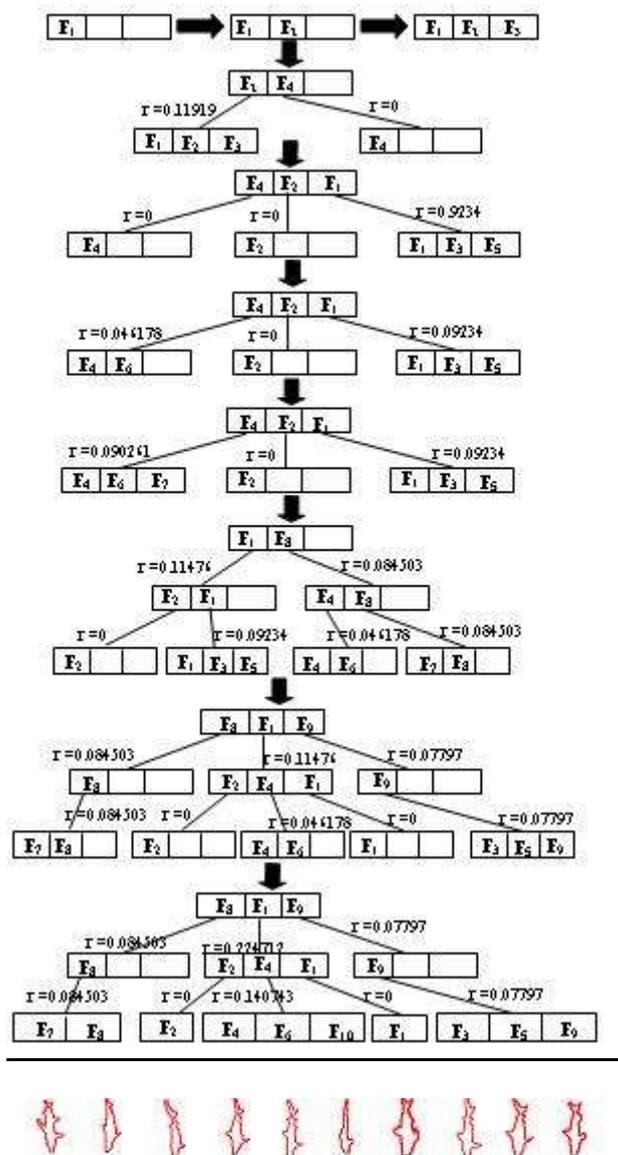


Figure 6. Construction de l'arbre M-tree

4.3. Recherche d'une forme requête

La recherche d'un objet requête se fait en parcourant la structure d'arbre en profondeur tout en éliminant le parcours des branches inutiles.

Nous avons mené plusieurs expérimentations. La figure 7 montre un exemple d'une forme requête et les résultats obtenus de la recherche des k-plus proches voisins en fixant k à 20. Les 20 premières formes les plus ressemblantes à la forme requête sont affichées de gauche à droite puis de haut en bas.

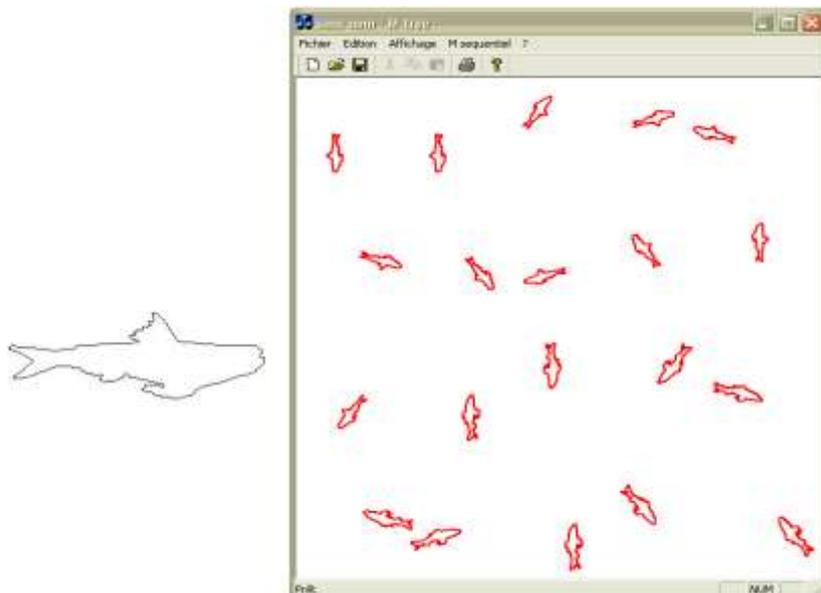


Figure 7. Forme requête et les 20 premiers résultats obtenus

Nous avons pu constater que les dix premiers résultats correspondent à la même forme requête dans différentes orientations. Ceci démontre l'efficacité de notre descripteur concernant l'invariance par rapport à la rotation. Notons que les résultats obtenus sont très proches de ce qu'un utilisateur aurait pu trouver visuellement. Les formes obtenues ont différentes orientations.

4.4. Temps de réponse

Nous avons mené également plusieurs tests pour montrer la performance de notre système en comparant notre méthode à la recherche séquentielle. La figure 8 présente le temps de réponse des deux méthodes en fonction de la taille de la base.

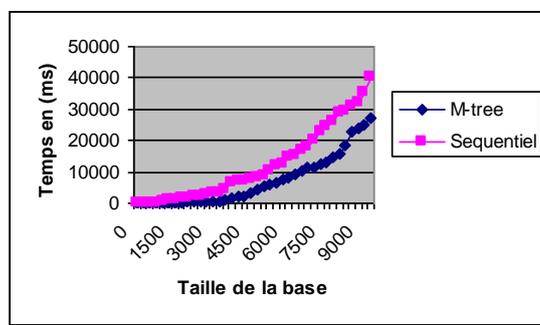


Figure 8. Comparaison entre M-tree et la méthode séquentielle

Les résultats obtenus permettent de montrer l'intérêt de notre approche au niveau de l'indexation de la base de formes en réduisant d'une façon remarquable le temps réponse.

5. Conclusion et perspectives

Dans cet article, nous avons proposé une nouvelle approche pour la description, l'indexation et la recherche dans les grandes bases de formes. Le descripteur ainsi proposé est une version multi-échelle de celui de Berretti et al., qui n'est pas invariant par rapport à la rotation. Pour le rendre ainsi, nous avons proposé de réaliser une rotation de tous les vecteurs d'orientations des tokens suivant un angle calculé en utilisant le principe d'équilibre des vecteurs des forces. De plus, les attributs d'un token sont calculés à plusieurs valeurs d'échelles. En ce qui concerne l'indexation, au lieu d'associer un arbre M-tree à chaque forme de la base, nous avons stocké l'ensemble des index de la totalité de la base de formes dans une seule structure en arbre M-tree. Ceci permet d'éviter le parcours séquentiel lors de la recherche dans la base. Les résultats obtenus, par l'application de notre système sur une grande base de formes, montrent l'intérêt des deux contributions de description et d'indexation de formes. Nous avons montré aussi que le système proposé est plus performant en temps de réponse que les autres systèmes basés sur le parcours séquentiel.

Parmi les perspectives, nous envisageons d'intégrer d'autres descripteurs de formes pour enrichir notre système. Tester d'autres techniques d'indexation pourra également, faire l'objet d'un travail ultérieur.

Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B. (1990). "The R*-tree: An efficient and robust access method for points and rectangles," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, pp.322–331.

Berretti, S., Del Bimbo, A., Pala, P. (2000). "Retrieval by shape Similarity with Perceptual Distance and Effective Indexing". *IEEE Transactions On Multimedia, Vol. 2, No 4*, pp 225-239.

Binaghi, E., Gagliardi, I., Schettini, R. (1994). "Image retrieval using fuzzy evaluation of color similarity," *Int. J. Pattern Recognit. Artif. Intell.*, vol.8, no. 4, pp. 945–968.

Del Bimbo, A. , Pala, P. (1997). "Visual image retrieval by elastic matching of user sketches," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 121–132.

Ciaccia, P., Patella, M., Rabitti, F., Zezula, P. (1997). "Indexing Metric Spaces with M-tree". In *SEBD '97*, pp 67-86.

Faloutsos, C., Flickner, M., Niblack, W., Petkovic, D. , Equitz , W., Barber , R. (1993). "The Qubic Project: Efficient and Effective Querying by Image Content," *IBM Res. Div. Almaden Res. Center*, Res. Rep. 9453.

Guttman, N. (1984), « R-trees : A dynamic index structure for spatial searching », In *Proc. 1984 ACM SIGMOD Int. Conf. Management of Data, Bostone, MA, June* , pp 47-57.

Liu, F., Picard, R. W. (1996). "Periodicity, directionality, and randomness —Wold features for image modeling and retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 722–733.

Mahmoudi, S., Daoudi, M. (2002). "Une nouvelle méthode d'indexation 3D", *13ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA2002), Angers, France, volume 1*, pp. 19-27.

Mokhtarian, F., Abbasi, S., Kittler, J. (1996). "Efficient and robust retrieval by shape content through curvature scale spac", *Proc. International Workshop on Image Databases and MultiMedia Search, Amsterdam, The Netherlands*, pp 35-42.

- Remolar, I., Chover, M., Belmonte, Ó., Ribelles, J., Rebollo, C. (2002). " *Geometric Simplification of Foliage*", *EUROGRAPHICS*.
- Sellis, T., Roussopoulos, N., Faloutsos, C. (1987). "The R+ -tree: A dynamic index for multi-dimensional objects," in *Proc. 13th VLDB Int. Conf.*, pp. 507–518.
- Tamura, H., Mori, S., Yamawaki, T. (1976). "Texture features corresponding to visual perception," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp.460–473.
- Ulmann, J. K. (1991). Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40(4):175-179 .
- Veltkamp, R.C., Tanase., M. (2000). Content-based image retrieval systems : A survey. *Technical Report, Department of Computing Science, Utrecht University*, UU-CS-2000-34.
- Zhang, D. , Lu. (2004). "Review of shape representation and description techniques", *Pattern Recognition* 37 1 – 19.