

Une nouvelle méthode de description et d'indexation des grandes bases de formes

H. SILKAN¹, S. E. OUATIK¹, A. LACHKAR², M. MEKNASSI¹

¹LISQ, Faculté des Sciences Dhar EL Mahraz, Fès

²E.S.T.M, Université Moulay Ismail, Meknès, lachkar@est-umi.ac.ma,
Silkan_h@hotmail.com, souatik@fsdmfes.ac.ma, m_meknassi@yahoo.com

Résumé

Dans ce papier, nous proposons une nouvelle méthode de description et d'indexation des grandes bases de formes. Elle apporte deux contributions. La première consiste à calculer un nouveau descripteur de forme invariant par rapport à certaines transformations géométriques, notamment 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. Dans la deuxième contribution, nous proposons 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 celle-ci. Nous montrerons aussi que notre système d'indexation et de recherche est plus performant en temps de réponse que les autres systèmes basés sur le parcours séquentiel.

Mots clefs

Descripteur invariant, Indexation, M-tree, Recherche par similarité, Bases de formes.

1 Introduction

La quantité d'images produites et archivées chaque jour ne cesse d'augmenter. Pour pouvoir exploiter ces archives colossales ainsi créées, il faut être capable d'indexer ce qui est stocké, et le retrouver. Ceci nécessite, d'une part, le choix d'une représentation pertinente des images de la base par le biais de primitives visuelles significatives et fiables qui traduisent le contenu de la base. D'autre part, une structure d'index efficace est nécessaire 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*, tels que la couleur [1,2] et la texture [3,4]. La forme [5,6] tient une place à part 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 [7]. Récemment, Berretti et al. [8] ont proposé un nouveau descripteur. Les formes sont partitionnées en des portions appelées *tokens*. Chaque token est représenté par deux attributs qui sont la *courbure* et l'*orientation*.

Comme il est mentionné par Zhang et al. [9], 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 à ce problème, nous proposons de réaliser une rotation de tous les vecteurs d'orientations des tokens suivant un angle calculé en utilisant le principe d'équilibre des vecteurs de forces. De plus, les deux attributs courbure et orientation sont calculés à plusieurs valeurs d'échelles. Il s'agit alors d'une version multi-échelle du descripteur de Berretti et al. [8], ayant l'avantage d'être invariant 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 se basant principalement sur deux étapes : (i) organiser les vecteurs caractéristiques dans des structures de données adaptées, puis (ii) trouver les algorithmes de parcours efficaces pour améliorer le temps de la recherche. Les méthodes d'accès spatial SAM (Spatial Access Methods) comme R-Tree [10] et ses variantes [11,12] ont été utilisées pour répondre aux requêtes de similarité. Il s'agit de retrouver les objets de la base les plus similaires à un objet requête. Ainsi, ce processus nécessite une fonction de distance pour mesurer la similarité entre les vecteurs caractéristiques. Cependant, l'application des méthodes SAM a relevé deux limitations (i) le nombre de composantes des vecteurs caractéristiques doit être réduit, (ii) la fonction de distance doit être de type L_p comme la distance euclidienne L_2 et Manhattan. Berretti et al. [8] ont utilisé l'arbre métrique M-tree [13]. 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 [14] proposent de caractériser les objets 3D par des vues caractéristiques. Ils ont utilisé des index calculés à partir des vues et basés 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. [8]. Chaque token est représenté par deux attributs : la courbure et l'orientation.

Soit $f(u) = \langle (x(u), y(u)) | u \in [0, T] \rangle$ 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 $\langle f(u, \sigma) | \sigma \geq 0 \rangle$ 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 $\langle f(u, \sigma) | \sigma \geq 0 \rangle$.

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

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°.

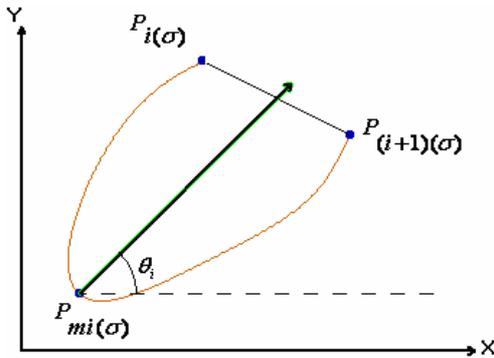


Figure 1 – Représentation d'un token : l'angle θ_i mesure l'orientation du token.

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

$\Gamma_\sigma = \langle E_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)}) | 1 \leq i \leq n_\sigma \rangle$, Où n_σ est le nombre de tokens de la forme lissée notée F_σ . Finalement le vecteur caractéristique obtenu pour une forme de la base est donné par :

$$\Gamma = \prod_{j=1}^M \Gamma_{\sigma_j} = \prod_{j=1}^M \langle E_{i(\sigma_j)}(m_{i(\sigma_j)}, O_{i(\sigma_j)}) | 1 \leq i \leq n_{\sigma_j} \rangle = \langle E_i(m_i, O_i) | 1 \leq i \leq n \rangle$$

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 invariants par rapport à la translation et à l'homothétie. Et par conséquent, notre descripteur sera également invariant par rapport à ces transformations. Le problème qui reste est celui d'invariance par rapport à la rotation. Prenons un exemple de deux orientations différentes d'un même token (figure-2).

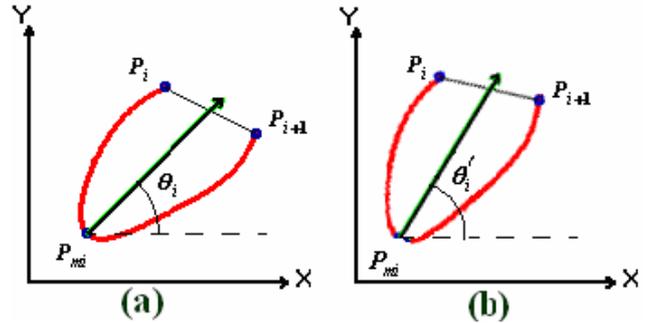


Figure 2 - Un token représenté dans deux orientations différentes.

Soient $E_1(m_i, \theta_i)$ et $E_2(m'_i, \theta'_i)$ 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 a_0 et b_0 deux vecteurs non nuls. D'après le principe d'équilibre des vecteurs de forces, il existe un vecteur c_0 tel que $a_0 + b_0 + c_0 = \vec{0}$. Le vecteur 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 a_θ et b_θ sont les deux vecteurs obtenus après la rotation de a_0 et b_0 par un angle θ . Il existe un vecteur principal c_θ qui vérifie $a_\theta + b_\theta + c_\theta = \vec{0}$. La direction principale θ de c_θ est déterminée par l'équation : $\theta = \arccos\left(\frac{c_0 \cdot c_\theta}{|c_0| \cdot |c_\theta|}\right)$

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 $\Lambda_0 = \{f_i^p | 1 \leq i \leq N\}$, où N est le nombre de vecteurs et $\Lambda_\theta = \{g_i^p | g_i^p = R(\theta) \cdot f_i^p, 1 \leq i \leq N\}$ sa rotation par un angle θ . Alors, si $\sum_{i=1}^N f_i^p \neq \vec{0}$, il existe un vecteur principal F_0 vérifiant $\sum_{i=1}^N f_i^p + F_0 = \vec{0}$. De même, si

$\sum_{i=1}^N g_i^p \neq \vec{0}$ il existe un vecteur principal F_θ vérifiant l'équation (1)

$$\sum_{i=1}^N g_i^p + F_\theta = \vec{0} \quad (1).$$

La direction principale θ de F_θ est donc déterminée par

$$\theta = \arccos\left(\frac{F_0 \cdot F_\theta}{|F_0| \cdot |F_\theta|}\right) \quad (2)$$

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

$$R(-\theta) \cdot \left[\sum_{i=1}^N g_i^p + F_\theta \right] = \sum_{i=1}^N [R(-\theta) \cdot g_i^p] + R(-\theta) \cdot F_\theta = \vec{0}$$

$$\text{Or, } R(-\theta) \cdot F_\theta = F_0 \quad \text{et} \quad F_0 = -\sum_{i=1}^N f_i^p$$

$$\text{Donc } \sum_{i=1}^N R(-\theta) \cdot g_i^p = \sum_{i=1}^N f_i^p \quad (3)$$

D'après la formule (3), on déduit que la direction initiale de l'ensemble $\Lambda_0 = \{f_i^p | 1 \leq i \leq N\}$ peut être déduite à partir de la rotation des vecteurs $\sum_{i=1}^N g_i^p$ et F_θ par

l'angle $-\theta$. Cela peut être réalisé par l'application des deux formules suivantes :

$$R(-\theta) \cdot \Lambda_\theta = \Lambda_0 \quad (4)$$

$$R(-\theta) \cdot F_\theta = F_0 \quad (5)$$

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 Λ_{θ_1} et Λ_{θ_2} d'une même forme avec deux orientations différentes, on calcule d'abord leurs vecteurs principaux respectivement F_{θ_1} F_{θ_2} . Ensuite, on détermine leurs directions principales θ_1 et θ_2 . Puis, on procède à une rotation de l'ensemble des vecteurs Λ_{θ_1} et Λ_{θ_2} respectivement par $-\theta_1$ et $-\theta_2$ en appliquant (4) et (5). On obtient alors pour Λ_{θ_1} et Λ_{θ_2} le même vecteur caractéristique Λ_0 ayant comme vecteur principal F_0 , vie 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).

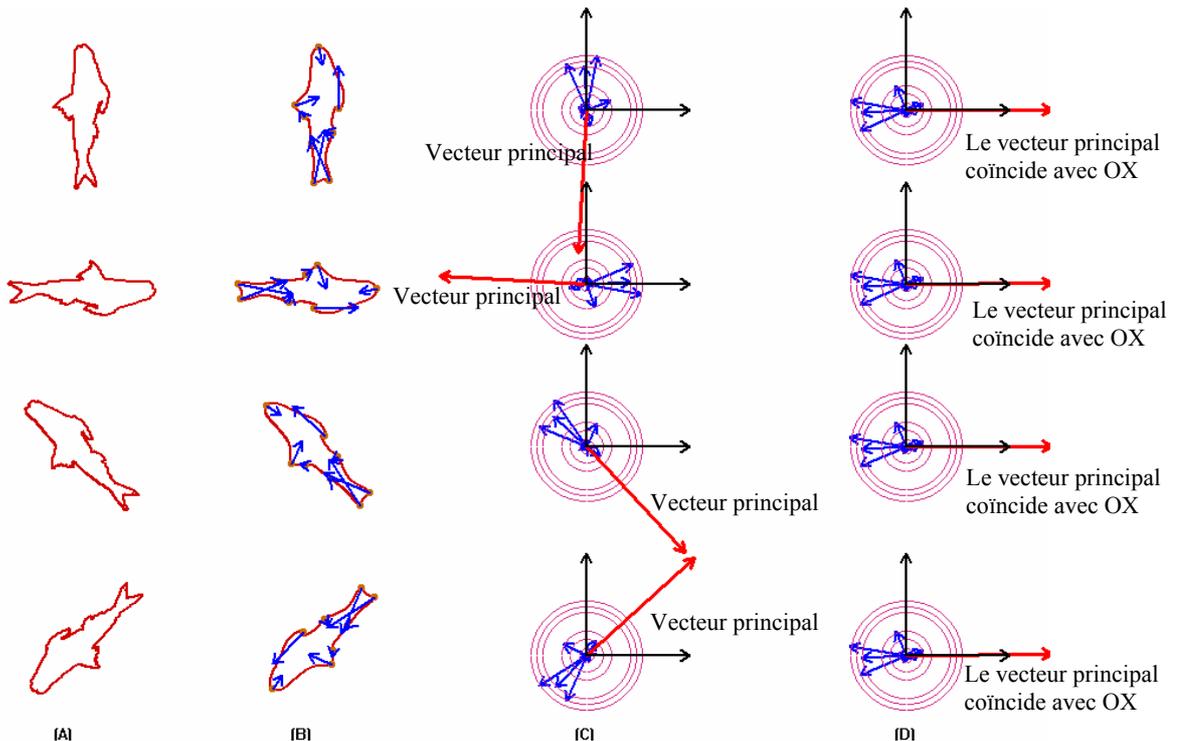


Figure 3 - Invariance par rapport à la rotation : (A) une même forme dans quatre orientations différentes. (B) forme avec les vecteurs caractéristiques associés. (C) présentation des vecteurs caractéristiques et le vecteur principal dans un repère euclidien. (D) présentation des différents vecteurs après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

Notre algorithme utilisé pour calculer le descripteur est le suivant :

```

Procédure calcul_descripteur(Forme F)
{ Soit A : l'ensemble des vecteurs (courbure, orientation)
  Soit Aθ : l'ensemble des vecteurs caractéristiques
  A ← ∅ ; Aθ ← ∅ ;
  Pour chaque valeur de lissage σ faire
  { Pour chaque token fi de F
    { Calculer Ei(σ)(mi(σ), Oi(σ)).
      Ajouter Ei(σ) à A.
    }
  }
  Calculer la direction principale θ de l'ensemble A.
  Pour chaque vecteur Ei de A faire
  { Ei' ← R(-θ).Ei
    Ajouter Ei' à Aθ
  }
}

```

Notons que notre descripteur présente plusieurs avantages :

- Le lissage permet d'avoir la robustesse au bruit.
- L'invariance par rapport à la translation, l'homothétie et la rotation.
- Il permet d'éviter le problème du choix du token de référence utilisé dans le travail de Berretti et al. [8].

3 Méthode d'indexation

Nous avons choisi une méthode d'indexation basée sur les arbres métriques [15], 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 :

$$d(O_x, O_y) = d(O_y, O_x) \quad (\text{symétrie})$$

$$d(O_x, O_y) > 0 \text{ si } O_x \neq O_y \text{ et } d(O_x, O_x) = 0 \quad (\text{Positivité})$$

$$d(O_x, O_y) \leq d(O_x, O_z) + d(O_z, O_y) \quad (\text{inégalité du triangle})$$

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 qui sont 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 du nœud :

- Entrée d'une feuille :
entry(F_j) = [F_j, oid(F_j), d(F_j; P(F_j))] où :
F_j : les descripteurs de la forme F_j.
oid(F_j) : l'identifiant de la forme (nom du fichier image).
d(F_j; P(F_j)) : distance entre F_j et P(F_j) qui est l'objet père .
- Entrée d'un nœud interne :
entry(F_r) = [F_r, ptr(T(F_r)), r(F_r), d(F_r; P(F_r))] où :
F_r : les descripteurs de la forme F_r (routing object).
ptr(T(F_r)) : pointeur sur un sous arbre T(F_r)

r(F_r) : le rayon de couverture, c'est la distance maximale entre F_r et les formes de la sous région.

d(F_r; P(F_r)) : distance entre F_r et P(F_r) qui est l'objet père. 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 (appel à 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, toute l'information 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).

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 pour mesurer la similarité entre deux formes [16].

Soient $X = \{x_i(m_i, O_i) / 1 \leq i \leq n\}$ l'ensemble de descripteurs d'une forme F et $Y = \{y_j(m_j, O_j) / 1 \leq j \leq m\}$

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) est la distance entre deux vecteurs donnée par :

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

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

4 Résultats

Pour illustrer l'intérêt de notre méthode, nous avons développé une application en Visual C++. Elle permet à l'utilisateur d'interroger la base de formes à partir d'une interface graphique. Nous avons utilisé une base de 1100 images d'animaux marins [17].

4.1 Construction de l'arbre

La construction de l'arbre est effectuée d'une manière

dynamique du bas vers le haut. Nous avons utilisé l'algorithme de construction présenté dans [13]. Chaque entrée d'un nœud de l'arbre stocke un ensemble de vecteurs caractéristiques correspondant à une forme de la base. La figure 5 montre un exemple de construction de l'arbre pour 10 formes de la figure 4.

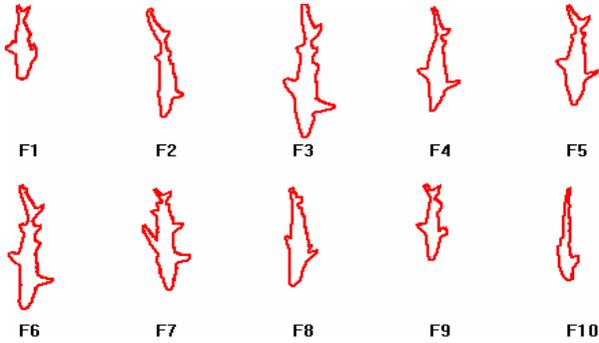


Figure 4 – Exemples de formes de la base

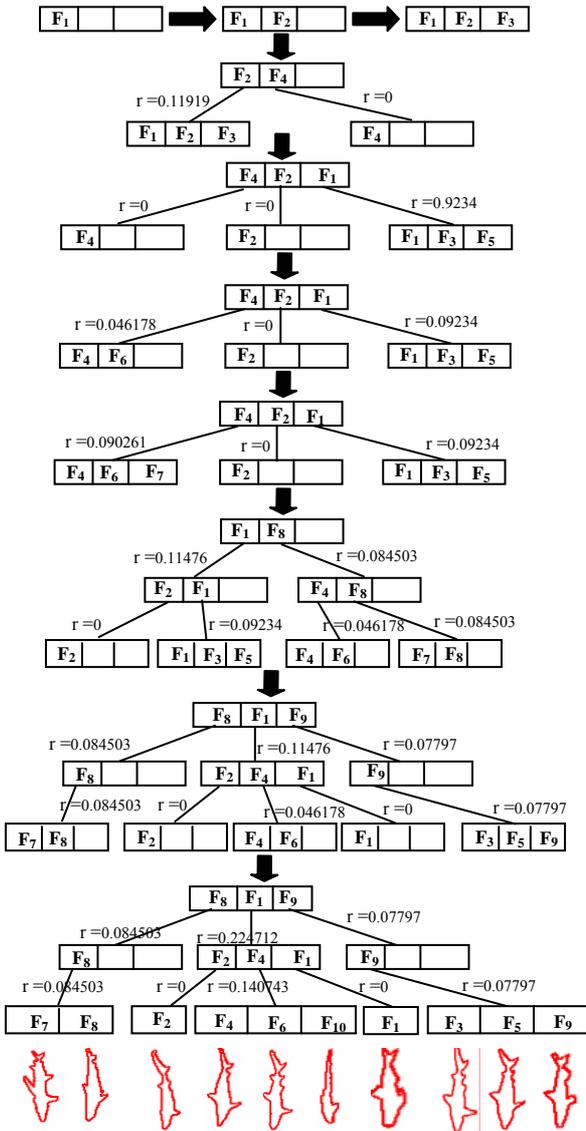


Figure 5 - Construction de l'arbre M-tree

4.2 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. Les figures 6 et 7 montrent chacune un exemple d'une forme requête et les résultats obtenus de la recherche des k-plus proches voisins en fixant k à 9. Les 9 premières formes les plus ressemblantes à la forme requête sont affichées de gauche à droite puis de haut en bas.

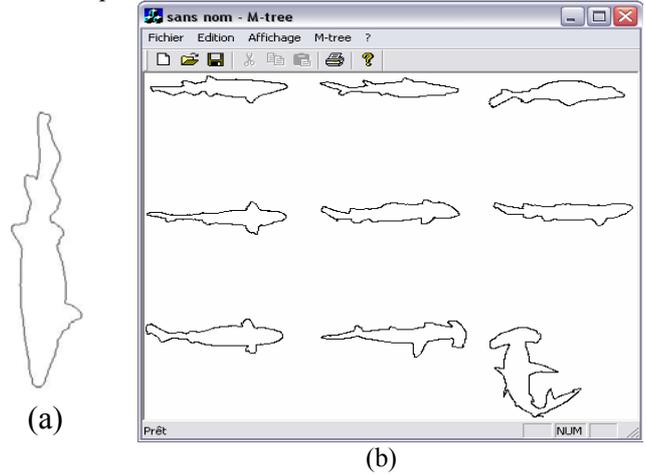


Figure 6 - (a) Forme requête (b) les plus proches voisins obtenus.

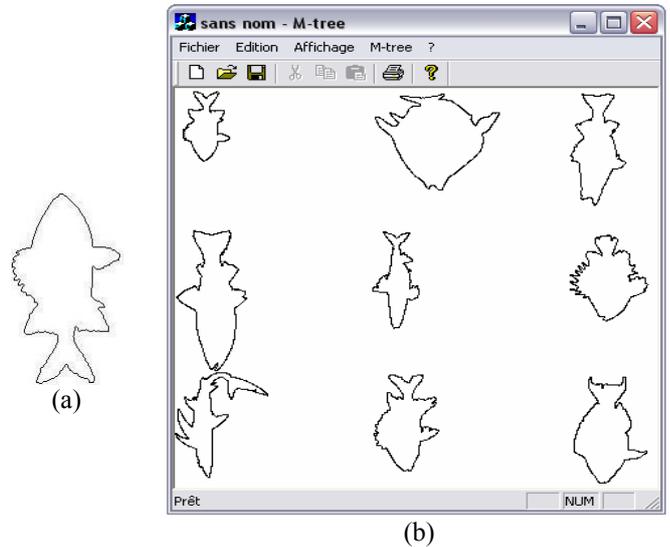


Figure 7 - (a) Forme requête (b) les 9 plus proches voisins obtenus.

Notons que les résultats obtenus sont très proches de ce qu'un utilisateur aurait pu trouver visuellement. Les formes obtenues ont différentes orientations.

4.3 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. 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.

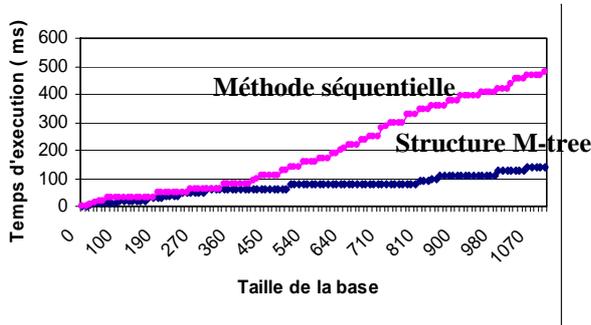


Figure 8 - Comparaison entre M-tree et la méthode séquentielle

5 Conclusion et perspectives

Dans ce papier, 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 réalisé 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. D'autres expérimentations menées sur une base de formes de taille plus grande (11 000 formes) ont été présentées dans un autre travail [18].

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

Références

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

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

[3] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness —Wold features for image modeling

and retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 722–733, July 1996.

[4] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 460–473, Apr. 1976.

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

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

[7] R.C. Veltkamp et M. Tanase. Content-based image retrieval systems : A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, October 2000.

[8] S. Berretti, A. Del Bimbo, P. Pala "Retrieval by shape Similarity with Perceptual Distance and Effective Indexing". *IEEE Transactions On Multimedia*, Vol. 2, No 4., December 2000

[9] Dengsheng Zhang, Guojun Lu., "Review of shape representation and description techniques", *Pattern Recognition* 37 (2004) 1 – 19.

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

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

[12] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The r+-tree: A dynamic index for multi-dimensional objects," in *Proc. 13th VLDB Int. Conf.*, Sept. 1987, pp. 507–518.

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

[14] S. Mahmoudi et M. Daoudi, "Une nouvelle méthode d'indexation 3D", 13ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA2002), volume 1, pp. 19-27, Angers, France, 8-9 janvier 2002.

[15] J. K. Ulmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40(4):175-179, November 1991.

[16] I. Remolar, M. Chover, Ó. Belmonte, J. Ribelles, C. Rebollo, "Geometric Simplification of Foliage", *EUROGRAPHICS 2002*.

[17] <http://www.ee.surrey.ac.uk/Research/VSSP/imagedb>.

[18] H. Silkan, A. Lachkar, S. E. Ouatik, A. Elkharraz, "Une nouvelle approche pour la description, l'indexation et la recherche dans les grandes bases de formes. Soumis à MCSEAI, Agadir December 7-9, 2006, Maroc.