

Recherche d'images à partir d'une requête partielle

utilisant la disposition des régions

Image retrieval from a partial request using region disposition

Sylvie Philipp-Foliguet, Mohamed Lekkat

Equipes Traitement des Images et du Signal,
CNRS 8056, ENSEA/UCP,
6 av. du Ponceau, 95014 Cergy Cedex, France
philipp@ensea.fr

Résumé

Le but de cet article est de pouvoir retrouver des images contenant un objet spécifique ou un groupe d'objets dans une base d'images. Cette recherche est basée sur un algorithme de mise en correspondance de graphes, adapté aux graphes d'adjacence de régions. Le nombre de régions représentant l'objet peut différer d'une image à l'autre. L'article propose d'une part une solution pour réduire la combinatoire de l'appariement de graphes et d'autre part une mesure de similarité entre graphes permettant d'ordonner les images résultats. L'adjacence et la relation spatiale entre les régions sont employées pour accélérer la recherche.

Mots Clef

Recherche d'image, appariement de graphes, région floue, relation spatiale

Abstract

The purpose of this paper is to retrieve images containing a specific object or group of objects in an image database. This search is based on an algorithm of graph matching, adapted to adjacency graphs of regions. The number of regions representing the object can differ from an image to the other one. The paper proposes on one hand a solution to reduce the combinatory of the graph matching and on the other hand a measure of similarity between graphs allowing to rank the retrieved images. The adjacency and the spatial relationships between regions are used to accelerate the search.

Keywords

Image retrieval, graph matching, fuzzy region, spatial relationship.

1. Introduction

En indexation et recherche d'image par le contenu, peu de travaux ont été menés sur la composition des images, c'est-à-dire sur l'agencement spatial des entités composant l'image. Ceci est sans doute en partie dû à la difficulté d'obtenir ces entités de manière automatique sur toute une base d'images. Pourtant en vision pré-attentive, notre système visuel perçoit grossièrement des zones avec leur couleur, leur forme, leur taille, et leur position dans l'image, et ceci relativement aux autres entités présentes dans l'image. Pour reconnaître un paysage, ou une peinture, notre système visuel n'effectue pas une segmentation précise de la scène, et pourtant cette reconnaissance est instantanée.

Ces dernières années, la segmentation a été revisitée pour obtenir des régions, même imprécises, mais sans avoir à régler des seuils pour chaque image [1, 3, 7, 9, 10, 14, 16]. Nous avons proposé un algorithme qui contourne le problème de la segmentation en représentant les régions comme des ensembles flous [11].

Après segmentation, qu'elle soit floue ou nette, les objets sont souvent divisés en plusieurs régions. Par ailleurs, lorsqu'on recherche une image, la requête peut être partielle, c'est-à-dire que l'utilisateur cherche une partie d'une image, par exemple un objet, une personne ou un ensemble d'objets, qui ne forme pas une entité homogène mais qui peut se représenter comme un ensemble de régions. La requête peut donc se formuler sous la forme d'un ensemble de régions, connexe ou non. La recherche consiste alors à retrouver dans la base, les images comprenant un ensemble de régions de mêmes caractéristiques et dans les mêmes positions relatives que l'ensemble des régions formant la requête.

Dans des systèmes de recherche basés sur les régions, la topologie des régions, au sens de leurs rapports spatiaux est rarement prise en compte. Dans des systèmes qui s'intéressent à la topologie, les images sont souvent représentées par des graphes, tels que les graphes d'adjacence. Le problème est alors de mettre en correspondance un ensemble de régions constituant la requête avec un autre ensemble de régions, extraites d'une image de la base. La requête peut être globale (ensemble de toutes les régions de l'image) ou partielle (partie d'une image, connexe ou non). Nous ne considérons pas de requête faite de régions provenant d'images différentes, qui n'ont alors aucun lien topologique.

À cause des variations d'échelle, de position, d'éclairage, etc., le nombre de régions composant la requête n'est pas nécessairement le même dans l'image requête et dans l'image cible.

Dans cet article nous nous focalisons sur la distance entre deux sous-graphes représentant un ensemble de régions requête et un ensemble de régions cibles, ces ensembles pouvant être de tailles différentes. Le problème est d'une part de réduire la combinatoire de la mise en correspondance de graphes et d'autre part de définir une mesure de similarité entre graphes appariés.

Pour être en mesure de trouver une image cible en un temps acceptable, nous proposons un algorithme qui écarte rapidement les images et les régions trop dissemblables aux requêtes et qui soit adapté à la mise en correspondance inexacte.

Nous proposons aussi une distance entre des sous-graphes, permettant un classement des images. Cette distance tient compte à la fois de la distance entre les attributs des régions et de la position relative de régions.

2. Algorithme de mise en correspondance

Nous nous intéressons à une requête constituée d'un ensemble de régions, adjacentes ou non. Le problème est de rechercher dans chaque image de la base l'ensemble des régions qui s'apparie au mieux à l'ensemble des régions requête.

Après segmentation en régions, chaque image de la base de données est représentée sous la forme d'un graphe valué (« attributed relational graph »). Les nœuds du graphe correspondent aux régions et les arêtes aux adjacences entre régions. Ce graphe n'est pas orienté.

La requête est un sous-graphe du graphe valué représentant l'image (appelée *image requête*), il est constitué d'une ou plusieurs composantes connexes. La recherche consiste alors à rechercher dans chaque graphe valué représentant une image de la base de données, le

sous-graphe qui correspond au mieux au sous-graphe requête.

2.1 Travaux antérieurs

Si on regarde la bibliographie en indexation d'images employant des régions, la composition de l'image n'est souvent simplement pas prise en compte comme dans [7] ou dans Blobword [1] et le score final de similarité pour un ensemble de régions est un opérateur simple comme le *min* ou le *max* des similarités entre les couples de régions des deux images. En général la topologie de l'image est caractérisée par un attribut de région, comme le centre de gravité. Dans le système Picasso [3], la recherche d'un ensemble de régions emploie un algorithme récursif qui met en correspondance une région requête avec un nœud de la pyramide multirésolution d'images segmentées. Une région requête peut ainsi être mise en correspondance avec une région de n'importe quelle résolution et comme une région à une résolution donnée est l'union des régions aux résolutions plus fines, ceci peut s'interpréter comme une mise en correspondance entre sous-graphes de cardinalités différentes. La distance entre deux régions tient compte de la distance entre les centres de gravité et la similarité entre deux images est la somme des similarités entre les régions correspondantes. Dans NeTra [9], la localisation spatiale d'une région est mesurée par deux rectangles (intérieur et extérieur), qui sont employés comme contrainte et réduisent la recherche, mais aucune relation explicite entre les régions, comme l'adjacence, n'est considérée. Dans SIMPLICITY [16], la requête est un ensemble de régions et la distance entre deux ensembles est la somme des distances entre les régions appariées, pondérées par un score de satisfaction attribué à l'appariement. La distance ne tient pas compte de l'information spatiale, mais la méthode permet la mise en correspondance d'une région d'une image à plusieurs régions de l'autre image.

Des méthodes de mise en correspondance de graphes ont été récemment proposées en reconnaissance de formes. Hong et Huang [8] construisent un graphe valué modèle à partir d'un ensemble de graphes valués exemples. Cette méthode est bien adaptée aux logos ou plus généralement à des objets soumis aux seules déformations rigides. Robles-Kelly et Hancock [11] proposent un algorithme de mise en correspondance inexacte qui prend seulement en compte les adjacences entre les nœuds, les nœuds eux-mêmes n'étant pas valués.

Dans SIMPLICITY, la mise en correspondance peut se faire avec un nombre de régions différentes entre la requête et la cible [16]. L'algorithme apparie d'abord les paires de régions les plus semblables et pondère la distance selon un degré de satisfaction, dépendant de l'aire des régions. Aucune relation entre régions n'est prise en

compte. Dans VisualSEEK [15], la requête n'est pas constituée de régions, mais d'un ensemble de motifs colorés. Pour chaque motif, une liste de régions candidates est construite et la combinatoire est ensuite réduite à l'aide des positions relatives et absolues des régions, comparées à celles des modèles. Nous proposons d'employer ces relations pour réduire la combinatoire pendant la mise en correspondance des régions.

L'algorithme de mise en correspondance que nous présentons dans la suite de cet article permet l'appariement d'un nœud d'un graphe avec plusieurs nœuds de l'autre graphe. Deux sous-graphes appariés ne sont donc pas nécessairement identiques en termes de nombre de nœuds, d'attributs des nœuds ou de nombre d'arêtes, mais seulement semblables pour les attributs des nœuds (au sens d'une mesure de similarité appropriée) et cohérents en ce qui concerne les arêtes. Comme nous nous intéressons aux relations spatiales entre les régions, la cohérence porte sur l'adjacence et la position spatiale relative (comme au-dessus, à gauche, etc.).

Cordella et al ont présenté un algorithme de mise en correspondance inexacte entre graphes valués [2] : deux graphes sont semblables s'ils sont isomorphes par un ensemble de transformations. Ces transformations sont la décomposition d'un nœud en un sous-graphe, la fusion d'un sous-graphe en un nœud et l'insertion ou la suppression d'une branche.

2.2 Mise en correspondance de graphes appliquée à la recherche d'image : élagage de l'arbre de recherche

Nous avons employé la même approche que Cordella et al. [2] dans la construction du graphe de recherche, qui est représenté par un arbre (Fig. 1) et nous proposons des solutions à élagage de l'arbre en exploitant la dissimilarité entre les régions et la cohérence topologique. Pour chaque image de la base (appelée *image cible*) les distances entre toutes les paires région-requête, région-cible sont calculées.

À chaque niveau de l'arbre, un nœud représente un appariement entre une région-requête et une région de l'image cible. Un nœud est développé au niveau inférieur seulement si la dissimilarité entre les deux régions composant le nœud est inférieure à un seuil donné τ . Par exemple dans la figure 1, les nœuds R_1 et S_2 sont suffisamment similaires, tandis que R_1 et S_1 sont trop différents. Un arc est ajouté à un chemin de l'arbre seulement s'il y a cohérence topologique entre les deux paires de régions.

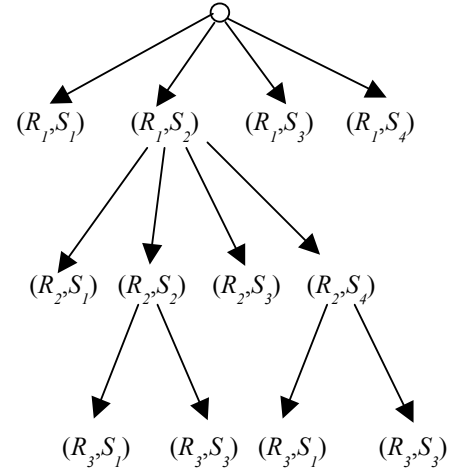


Figure 1 Arbre de recherche : chaque nœud correspond à un appariement de deux régions (R_i : région requête, S_j : région cible), chaque flèche correspond à une compatibilité topologique

La cohérence entre (R_i, S_j) et (R_{i+1}, S_k) tient compte de :

- l'adjacence de (R_i, R_{i+1}) et (S_j, S_k) qui doit être du même type (adjacent ou non)
- la position relative de (R_i, R_{i+1}) et (S_j, S_k) , par exemple si R_i est au-dessus de R_{i+1} , S_j doit être au-dessus de S_k .

Un appariement entre deux sous-graphes correspond à un chemin de la racine à une feuille de l'arbre de recherche. La dissimilarité entre sous-graphes est présentée dans le paragraphe 4. Comme c'est une somme des dissimilarités des nœuds d'un chemin, il est intéressant de mettre aux niveaux les plus hauts de l'arbre les couples de régions très dissemblables, l'arbre sera ainsi plus rapidement élagué.

Toutes les distances $d_{ij} = d(R_i, S_j)$ sont calculées et classées par ordre décroissant. Les couples (R_i, S_j) pour lesquels d_{ij} est supérieur à τ sont supprimés. Les couples restants sont stockés dans la liste ordonnée L , du moins semblable au plus semblable. Les régions requête sont placées dans l'arbre selon leur ordre dans cette liste. La région requête du premier couple (c'est-à-dire la région la moins semblable à n'importe quelle région cible) est mise au premier niveau de l'arbre (appelons-la R_1). On place au premier niveau de l'arbre tous les couples de L contenant R_1 . Au deuxième niveau de l'arbre on place le premier couple de L ayant une région requête différente de R_1 , avec autant de nœuds que cette région (appelons-la R_2) apparaît dans la liste, etc.

Par exemple l'arbre de la Fig 1 correspond à la liste suivante L (après seuillage) :

$$L = \{(R_1, S_3), (R_2, S_3), (R_1, S_4), (R_3, S_3), (R_1, S_1), (R_2, S_1), (R_2, S_4), (R_3, S_1), (R_2, S_2), (R_1, S_2)\}$$

Les propriétés de cet arbre sont les suivantes (deux régions sont dites similaires si leur dissimilarité est inférieure à τ :

- Une image qui n'a aucune région similaire avec une région de l'image requête est immédiatement écartée.
- La région requête qui correspond le moins avec l'image cible est examinée en premier
- Il y a au maximum autant de niveaux dans l'arbre que de régions requêtes. Il y a un niveau pour chaque région requête ayant au moins une région cible similaire.
- Dès que la dissimilarité globale entre deux graphes dépasse une valeur donnée, le nœud en cours n'est plus développé.

Chaque chemin de la racine à une feuille du dernier niveau représente un appariement de sous-graphes acceptable. Par exemple dans la figure 1, $\{(R_1, S_2), (R_2, S_2), (R_3, S_1)\}$ est l'un des quatre appariements de graphes possibles. On remarque qu'une région de l'image cible peut être appariée à plusieurs régions requête, mais pas l'inverse. On peut donc gérer une sous-segmentation de l'image cible par rapport à l'image de requête (Fig. 3a), mais pas une sur-segmentation (Fig. 3b).

Après cette première étape de construction de l'arbre, une deuxième étape consiste donc à tenter d'apparier plusieurs régions cibles avec la même région requête. Ceci est nécessaire si la segmentation a produit pour une seule région requête plusieurs régions dans l'image cible (par exemple Fig. 3b la région R_3 correspond aux régions S_1 et S_3). On examine les voisins de chaque région cible appariée et on recherche les régions susceptibles d'être appariées avec la région requête. Plus précisément, pour chaque appariement (R_i, S_j) , tous les voisins S_k de S_j sont examinés. Si $d(R_i, S_k)$ est inférieure au seuil τ et si l'appariement (R_i, S_k) est compatible avec les couples déjà appariés, alors R_i est mise en correspondance avec S_k . Dans l'exemple des figures 2 et 3b, R_3 est appariée avec S_1 ; S_1 et S_3 sont adjacentes, R_3 et S_3 sont suffisamment semblables et (R_3, S_3) est compatible avec (R_1, S_2) et (R_2, S_2) , alors R_3 est aussi mise en correspondance avec S_3 .

Un appariement entre graphes valués est finalement un ensemble de couples région-requête, région-cible suffisamment semblables et dont les adjacences et les positions relatives sont les mêmes dans les deux sous-graphes.

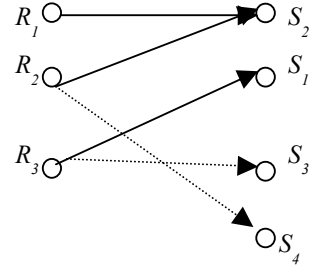
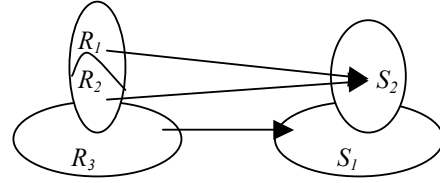
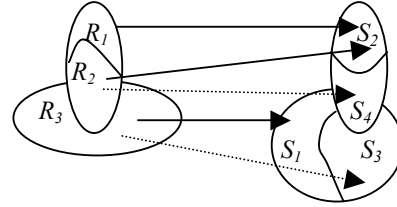


Figure 2 Mise en correspondance initiale, correspondant à un chemin de la Fig 1 (flèches continues), complété par la deuxième étape (flèches en pointillé).



(a) Exemple d'une image cible sous-segmentée, correspondant à l'appariement initial de la figure 2



(b) Exemple d'une image cible sur-segmentée correspondant à l'appariement complet de la figure 2 (flèches continues : première étape, flèches pointillées : deuxième étape)

Figure 3 Deux exemples d'appariements de graphes

3. Signature basée sur les régions floues

Pour obtenir une segmentation automatique, nous avons proposé un algorithme qui effectue une segmentation floue [11]. Une segmentation floue est un ensemble de régions floues, qui peuvent se chevaucher et dont l'union couvre l'image entière. Chaque région floue est un ensemble flou, caractérisé par les degrés d'appartenance des pixels, compris entre 0 et 1. Les degrés d'appartenance sont élevés quand le gradient est faible, c'est-à-dire dans des zones homogènes et diminuent sur les contours des zones, c'est-à-dire quand le gradient est élevé (voir figure 4).

Aucune normalisation n'est imposée (la somme des degrés d'appartenance d'un pixel à toutes les régions n'est pas nécessairement égale à 1), en accord avec [6].

Pour segmenter automatiquement une base d'images, il suffit de fixer le nombre approximatif de régions souhaité, par exemple entre 15 et 30 pour une base généraliste.

Un exemple est donné figure 4. La limite entre l'eau et la berge, ou l'eau et l'ombre de l'ours sont très franches, la région floue correspondante s'arrête nettement sur les bords. Au contraire, la limite de l'ours se confond parfois avec l'herbe, ou avec le sol, la région floue s'étale davantage, ses contours sont moins nets.



Une image de notre base de données généraliste

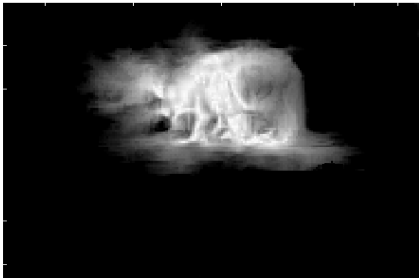


Figure 4 Deux régions floues obtenues à partir de l'image du haut ; au milieu, région correspondant à l'ours ; en bas, région correspondant à une partie de l'eau (plus c'est clair, plus le degré d'appartenance est élevé)

Les attributs de régions floues sont calculés en pondérant la contribution de chaque pixel s par son degré d'appartenance $\mu(s)$ à la région R [13]. Des attributs nets, géométriques ou colorimétriques peuvent ainsi être calculés [11]. Nous utiliserons le centre de gravité G de la région R , défini par ses coordonnées :

$$x_G = \frac{\sum_{s \in R} \mu(s) \cdot x(s)}{\sum_{s \in R} \mu(s)} \quad (1)$$

$$y_G = \frac{\sum_{s \in R} \mu(s) \cdot y(s)}{\sum_{s \in R} \mu(s)}$$

où $x(s)$ et $y(s)$ sont les coordonnées spatiales du pixel s .

La distribution de l'attribut a pour une région floue R définie par sa fonction d'appartenance μ a été définie par Dubois et Jaulent [4]. La probabilité pour tout réel r est définie par :

$$P_a(r) = \sum_{s \in R, a(s)=r} \mu(s) \quad (2)$$

La distribution colorimétrique est ainsi obtenue en ajoutant pour chaque classe couleur les degrés d'appartenance des pixels appartenant à cette classe, puis en normalisant. De cette façon les pixels avec des degrés d'appartenance faibles - appartenant aux transitions ou atypiques dans une région - ont peu d'influence sur la forme de distribution.

Les classes de couleur sont obtenues par l'algorithme des *k-means* adaptatif, calculé sur tous les pixels de la base, à partir de leurs composantes RVB. Le nombre de classes est de 25 [5].

Chaque image de la base de données est indexée par une signature, qui est composée de deux parties : d'une part l'ensemble des attributs de toutes les régions floues et d'autre part une représentation des relations topologiques entre les régions. Dans les résultats des paragraphes suivants, les régions sont décrites par leur seule distribution couleur (un vecteur de k valeurs par région) et les coordonnées du centre de gravité. Ces coordonnées sont normalisées en les divisant respectivement par la hauteur et la largeur de l'image ; les coordonnées sont ainsi comprises entre 0 et 1.

Le moyen le plus simple de représenter l'adjacence est d'employer la matrice d'adjacence des régions qui prend la valeur 1 si les deux régions ont au moins un pixel en commun, 0 sinon. Une solution plus précise serait de stocker le degré de chevauchement de régions. Elle n'a pas encore été testée.

La position relative des régions est prise en compte au travers des positions relatives de leurs centres de gravité lors de la construction de l'arbre.

4. Similarité entre régions et entre sous-graphes

4.1. Distance entre régions

Il y a de nombreuses mesures de dissimilarité entre distributions, nous avons employé une mesure simple tirée de la distance L_1 . Pour deux régions R et S respectivement représentées par les vecteurs couleur $(R^c, c = 1, \dots, k)$ et $(S^c, c = 1, \dots, k)$ la distance employée est :

$$d(R, S) = \frac{1}{2} \sum_{c=1}^k |R^c - S^c| \quad (3)$$

La normalisation est assurée par la division par 2. Cette distance est maximale et vaut 1 quand les deux distributions contiennent la seule valeur 1, placée en deux classes différentes.

4.2. Dissimilarité entre sous-graphes

Une fonction de dissimilarité f est associée à chaque chemin de la racine vers une feuille de l'arbre, qui correspond à un appariement initial de sous-graphes.

Plusieurs fonctions peuvent être employées. Un appariement initial M correspond à un chemin dans l'arbre de recherche comportant m nœuds, si m est le nombre de régions requête. M est donc un ensemble de m couples région-requête région-cible appariées :

$M = \{ (R_1, S^1), (R_2, S^2), \dots, (R_m, S^m) \}$, où R_i est une région requête, et S^i la région-cible appariée.

Si d représente une dissimilarité entre régions (par exemple celle de l'Eq 3), la dissimilarité entre sous-graphes, représentés par M est :

$$f(M) = \frac{1}{m} \sum_{i=1}^m d(R_i, S^i) \quad (4)$$

Quelle que soit la mesure de dissimilarité entre régions employée, comme on a toujours $d(R_i, S^i) \geq 0$, la fonction f croît au fur et à mesure qu'on ajoute des appariements de régions et qu'on prolonge les branches de l'arbre.

La deuxième partie de l'algorithme ajoute des appariements de régions. La fonction de dissimilarité pour un appariement de sous-graphe comprenant n ($n \geq m$) appariements de régions est :

$$f(M) = \frac{1}{n} \sum_{i=1}^n d(R_i, S^i) \quad (5)$$

4.3. Dissimilarité entre images

Finalement, la mesure de dissimilarité pour chaque image cible est le minimum des dissimilarités pour tous les appariements possibles de sous-graphes de l'image cible avec le sous-graphe formant la requête.

5. Résultats

Dans les tests suivants, nous n'avons utilisé que la cohérence verticale. Nous pensons que pour des bases de données généralistes, composées de paysages, de personnes, d'animaux, etc. seule la position verticale est importante pour retrouver des images semblables. Une symétrie par rapport à un axe vertical ne change pas notre perception de l'image, contrairement à une symétrie par rapport à un axe horizontal.

Nous avons utilisé une première base de données de 3000 images, automatiquement segmentées en régions floues. Chaque image est composée de deux objets de la base de données Colombia [17] qui contient 100 objets différents. Le problème est de retrouver des images contenant un objet quelle que soit sa position dans l'image. La Figure 5 montre un résultat avec une requête faite des trois régions floues constituant la tasse du haut de l'image. Toutes les images retrouvées contiennent cette tasse malgré des changements d'orientations et de petites variations d'échelle. La recherche est exécutée avec l'algorithme complet, permettant l'appariement d'une région requête avec plusieurs régions de l'image cible.

Les autres résultats ont été obtenus avec une base de données généraliste de 1200 images automatiquement segmentées (cf. figure 4).

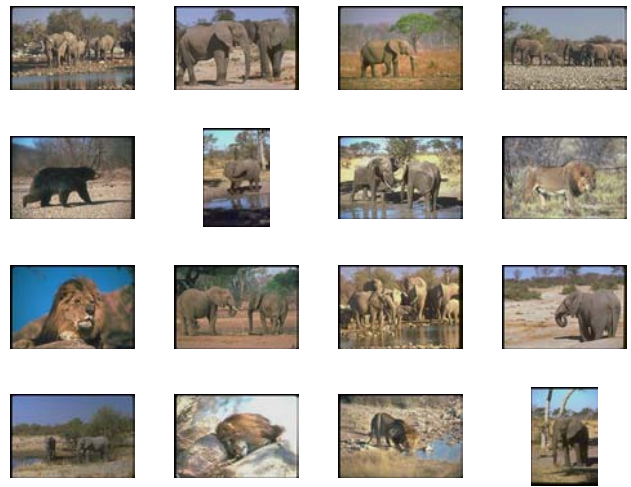


Figure 5 La requête est constituée de trois régions formant la tasse. Les images sont classées par dissimilarité croissante de haut en bas et de gauche à droite.

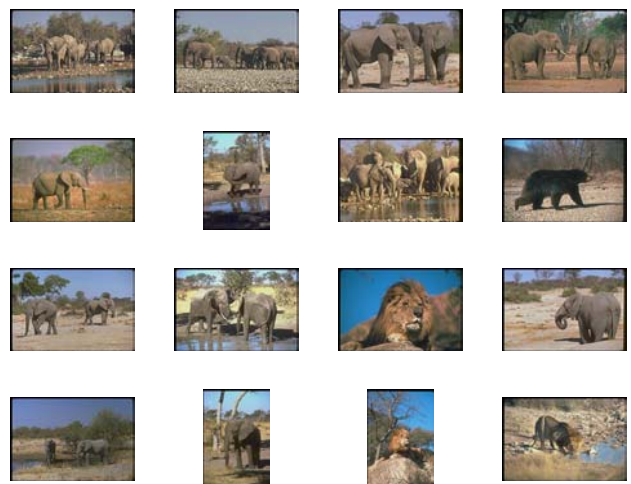
Nous comparons d'abord la version incomplète de l'algorithme (une région requête correspondant à une seule région de l'image cible) avec la version complète (une région requête pouvant être appariée à plusieurs régions de l'image cible). La figure 6 montre deux résultats, l'un avec la première partie de l'algorithme (une région cible pouvant être appariée à plusieurs régions requête mais pas l'inverse) et l'autre avec les deux possibilités d'appariement multiple. Il est net sur cet exemple, mais ceci a été confirmé dans tous les cas, que la mise en correspondance multiple dans les deux directions est une amélioration. On s'affranchit ainsi de l'imperfection de la segmentation, et on peut retrouver des images, ou des objets même si la segmentation est assez différente en nombre de régions d'une image à l'autre.

Nous avons ensuite étudié le lien entre le nombre de régions-requête et les résultats. Par exemple, il est plus facile de retrouver un avion en vol si on recherche également une région voisine bleue.

La figure 7 présente des résultats pour une recherche d'image d'éléphant. La base de données contient 14 catégories et 96 images d'« éléphants ». Les courbes de précision/rappel ont été obtenues après moyennage sur 5 requêtes à partir d'images d'éléphants. Les courbes sont obtenues successivement avec une requête faite de 1, 2 ou 3 régions prises dans les éléphants et de 4 régions (3 dans les éléphants et 1 dans le fond ou 2 dans les éléphants et 1 dans le fond).



a. Première partie de l'algorithme : une région requête correspond à une seule région de l'image cible. 11 images correctes sur 16, dont les 4 premières



b. Algorithme complet : appariements d'une région à plusieurs régions. 12 images correctes sur 16, dont les 7 premières

Figure 6 Requête partielle faite de 4 régions de l'image en haut à gauche (images classées de haut en bas et de gauche à droite)

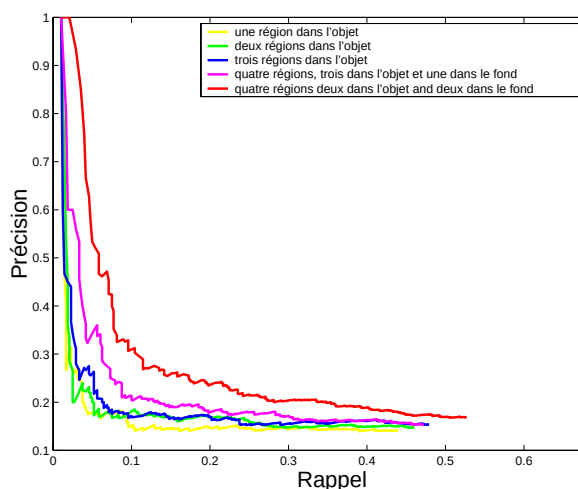


Figure 7 Courbes de rappel / précision pour 1 à 4 régions prises soit dans l'«objet» soit dans le fond.

On voit bien au travers de ce résultat qu'une requête faite de plusieurs régions, certaines dans l'«objet» et d'autres dans le fond donne de meilleurs résultats qu'une requête faite d'une partie seulement de l'«objet».

Dans la figure 8, le problème est de retrouver un ours brun près d'une étendue d'eau. La requête est constituée de deux régions recouvrant une partie de l'ours et une partie de l'eau (les deux régions présentées fig. 4). La figure 8 présente les images ordonnées. On obtient 12 images d'un ours au bord de l'eau parmi les 16 premières images retrouvées par le système. Toutes les images ne sont pas pertinentes, mais il faut noter que seule la distribution colorimétrique est prise en compte et qu'il n'y a pas de bouclage de pertinence. Figure 9 sont représentées les régions floues (au nombre de deux dans chaque cas) pour les deux meilleures images. On note que les deux régions les plus proches sont bien des régions appartenant à l'ours et à de l'eau dans les deux cas. Les régions appariées ont des formes différentes des régions requête, elles ont des tailles parfois bien différentes et ne sont pas situées au même endroit dans les images (la seule contrainte étant que la région «eau» doit être au-dessous de la région «ours»). On voit sur cet exemple l'intérêt de la segmentation floue, qui permet une grande souplesse dans la forme des régions.

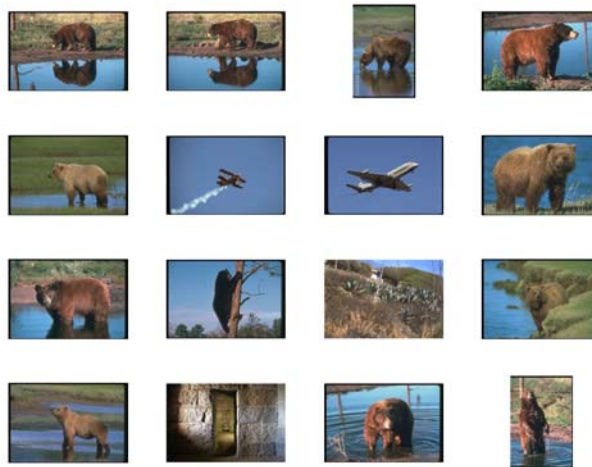


Figure 8 Résultat à partir d'une requête constituée de deux régions (l'ours et une partie de l'eau, présentées figure 4). Images classées par similarité décroissante. 12 images correctes dans les 16 premières.

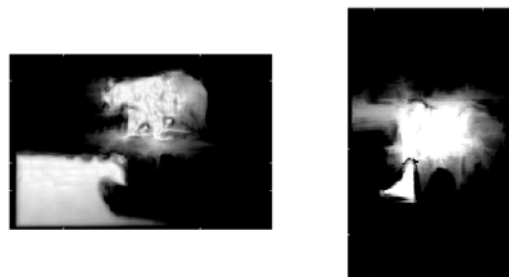


Figure 9 Les régions floues les mieux appariées aux deux régions floues requête affichées Fig. 4 pour les deux meilleures images résultat (images 2 et 3 de la Fig. 8)

6. Conclusion et perspectives

Nous avons développé un algorithme de mise en correspondance, adapté à la recherche d'image à partir d'une requête partielle, constituée de plusieurs régions d'une image.

Nous avons proposé des solutions pour réduire la combinatoire de mise en correspondance de graphes. Si le nombre de régions formant la requête est égal à n et le nombre de régions dans l'image cible est égal à p , le nombre d'appariements possibles de régions est $p \times n$ et le nombre maximum de chemins dans l'arbre est p^n . Lors de la deuxième étape de l'algorithme, pour chacune des p régions cible, il faut rechercher au maximum $p - 1$ régions

adjacentes et vérifier la compatibilité avec les couples déjà appariés. Le seuillage sur les dissimilarités entre régions réduit considérablement ce nombre (par exemple 90 % des couples peuvent ainsi être supprimés). L'utilisation de la cohérence d'adjacence réduit de nouveau ce nombre et la cohérence topologique limite finalement le nombre de régions appariées pour chaque image à un nombre en général compris entre $n/2$ et $n \times 2$. Comme le nombre de régions requête est faible (inférieur à cinq), le temps de traitement est compatible avec une utilisation en temps réel.

Cet algorithme peut être employé avec n'importe quelle base d'images segmentées, mais il a l'avantage de gérer des segmentations imprécises, puisqu'une région d'une des images peut être mise en correspondance avec plusieurs régions de l'autre image.

L'algorithme tient compte à la fois de la similarité entre les régions de la requête et de l'image cible et des positions relatives des régions dans chaque image. L'utilisation d'informations sur l'environnement d'un objet ou d'une personne peut être intéressante pour retrouver un type d'image (des éléphants sont en général dans la savane et les avions dans le ciel !).

Nous avons présenté des résultats avec un ensemble d'attributs simple (la distribution colorimétrique et la position des centres de gravité) et une distance simple. Les relations spatiales entre régions que nous avons utilisées se limitent à l'adjacence et à la position relative des centres de gravité de région. D'autres attributs (comme la texture) et la caractérisation plus sophistiquée des positions relatives des régions dans l'image amélioreront probablement les résultats. D'autres mesures de similarité, pondérant les attributs des régions sont à l'étude. On peut envisager par exemple une pondération par la taille de la région ou par des attributs topologiques entre régions plus précis.

Références

- [1] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld : image segmentation using Expectation-Maximisation and its application to image querying, *IEEE Trans. on PAMI*, 24 (8), 1026-1038, 2002
- [2] L. P. Cordella; P. Foggia, C. Sansone, M. Vento, Subgraph Transformation for the inexact Matching of Attributed Relational Graphs, *Computing*, 12, 43-52, 1998.
- [3] A. Del Bimbo, M. Mugnaini, P. Pala, F. Turco, Visual Querying by color perceptive regions, *Pattern Recognition*, 31 (9), 1241-1253, 1998
- [4] D. Dubois, M.C. Jaulent, A general approach to parameter evaluation in fuzzy digital pictures, *Pattern Recognition Letters* 6, 251-259, 1987
- [5] J. Fournier Indexation d'images par le contenu et recherche interactive dans les bases généralistes, PhD thesis, 2002, ETIS, Cergy, France, <http://www-etis.ensea.fr/~fournier>
- [6] R. Krishnapuram, J. M. Keller, A probabilistic approach to clustering, *IEEE Trans. on Fuzzy Systems*, 1 (2), 98-110, 1993
- [7] I. Kompatsiris, E. Triantafillou, G. Strintzis, Region-based color image indexing and retrieval, *IEEE ICIP*, Thessaloniki, Greece, 658-661, 2001
- [8] P. Hong, T. S. Huang, Spatial pattern discovering by learning the isomorphic subgraph from multiple attributed relation graphs, *Electronic Notes in Theoretical Computer Science*, 46, Elsevier Science Publishers, Sébastien Fourey, Gabor T. Herman and T. Yung Kong ed, 2001
- [9] W. Y. Ma, B. S. Manjunath, NeTra : a toolbox for navigating large image databases, *ACM Multimedia Systems*, 7 (3), 184-198, 1999
- [10] E.J. Pauwels, G. Frederix, Finding Salient Regions in Images: Non-Parametric Clustering for Image Segmentation and Grouping, *Journal of Computer Vision and Image Understanding*, 75, 73-85, 1999
- [11] S. Philipp-Foliguet, M. B. Vieira, A. de A. Araújo, Segmentation into fuzzy regions using topographic distance, 14th SIBGRAP, 282-288, Florianopolis, Brazil, 2001
- [12] A. Robles-Kelly, E. R. Hancock, Graph matching using adjacency matrix Markov chains, *Proc. 3rd Int. Workshop EMMCVPR*, Sophia-Antipolis, France, 2001
- [13] A. Rosenfeld, The fuzzy geometry of image subsets. *Pattern Recognition Letters* 2, 311-317, 1984.
- [14] A.W.M Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-Based Image Retrieval at the End of the Early Years, *IEEE Trans on PAMI*, 22 (12), 1349-1380, 2000
- [15] J. R. Smith, S-F. Chang, VisualSEEk : a fully automated content-based image query system, *ACM Multimedia Conf.*, Boston, USA, 87-98, 1996
- [16] J. Z. Wang, J. Li, G. Wiederhold, SIMPLcity : semantics-sensitive integrated matching for picture libraries, *IEEE Trans. on PAMI*, 23 (9), 1-17, 2001
- [17] <http://www.cs.columbia.edu/CAVE/research/softlib/coil-100.html>