

Recherche d'images par appariement d'ensembles de régions floues

Sylvie Philipp-Foliguet, Marcelo Bernardes Vieira,
Mohamed Lekkat

Equipes Traitement des Images et du Signal,
CNRS 8051, ENSEA/UCP,
6 av. du Ponceau, 95014 Cergy Cedex, France
philipp@ensea.fr

Résumé

Dans cet article nous présentons une méthode d'indexation et de recherche d'images dans une base qui prend en compte la disposition des régions dans l'image. L'indexation des images est basée sur une partition en régions floues et nous proposons un algorithme qui effectue une segmentation floue des images. La signature des images est constituée des attributs de régions floues et de la matrice d'adjacence des régions. La recherche d'image emploie un algorithme de mise en correspondance inexact de graphes. Nous proposons 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 exploitées pour accélérer la recherche. La méthode est adaptée aux requêtes partielles, elle permet par exemple de retrouver des images contenant un objet spécifique ou un groupe d'objets dans une base d'images.

Mots-clés : Indexation d'image, recherche d'image, appariement de graphes, segmentation floue, relation spatiale, requête partielle.

Abstract

In this paper we present a method of image indexing and retrieval in a base which takes into account the relative positions of regions in the image. The image indexing is based on a partition in fuzzy regions and we propose an algorithm which produces a fuzzy segmen-

tation of images. The signature of images is composed of the regions features and of the region adjacency matrix. The image retrieval is based on an inexact graph matching algorithm. We propose 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 result images. Adjacency and spatial relations between regions are used to accelerate the process. The method is adapted to partial requests, it allows for example to retrieve images containing a specific object or a group of objects in an image database.

Keywords : Image indexing, image retrieval, graph matching, fuzzy segmentation, spatial relationship, partial request.

1. INTRODUCTION

La recherche d'images par le contenu a connu un grand essor ces dernières années à cause de l'explosion du nombre d'images créées. Accéder aux images stockées dans des bases est devenu un problème majeur. Les buts toutefois sont très variés et nous abordons dans cet article un problème particulier, celui de la recherche d'images contenant un objet ou un animal donné ou placé dans un contexte particulier. Nous nous intéressons à une partie seulement de l'image, c'est pourquoi nous nous sommes tournés vers des représentations de l'image sous forme de régions.

La segmentation a été revisitée ces dernières années pour obtenir des régions, même imprécises, mais de manière automatique pour toute une base, c'est-à-dire sans avoir à régler des paramètres pour chaque image. 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 formes présentes dans l'image. Pour reconnaître un paysage, ou une peinture, notre système visuel n'effectue pas une segmentation précise de la scène, et néanmoins cette reconnaissance est instantanée. Nous avons cherché à reproduire ce mécanisme de vision imprécise en élaborant un algorithme de segmentation en régions floues.

Après segmentation, une entité sémantique peut se trouver partagée en plusieurs régions. Par ailleurs l'objet de la recherche ne forme pas nécessairement une entité homogène mais peut être représentée par un ensemble de régions. La requête se formule alors comme un ensemble de ré-

gions, connexes ou non, pour lesquelles les caractéristiques de couleur et de texture ainsi que les positions relatives vont être discriminantes.

Lorsque l'on s'intéresse à la topologie relative des régions, la représentation de l'image la plus utilisée est celle d'un graphe d'adjacence des régions. Les autres primitives souvent employées sont les points d'intérêt. Cependant ils ne nous ont pas paru adaptés à notre problème, d'abord parce qu'un animal par exemple présente une grande variabilité d'aspect et les points d'intérêt extraits sont donc très variables d'une image à l'autre. De plus les points d'intérêt sont en général situés sur les frontières et le plus souvent sur les angles des objets, donc les signatures calculées sur leurs voisinages tiennent compte à la fois des pixels à l'intérieur des objets et des pixels à l'extérieur des objets. Enfin le nombre de points d'intérêt calculés sur un objet est beaucoup plus important que le nombre de régions constituant l'objet. Le positionnement relatif des primitives est ainsi plus facile à modéliser avec des régions qu'avec des points d'intérêt.

Avec une représentation de l'image segmentée sous forme de graphe, la recherche d'objets consiste à mettre en correspondance le sous-graphe constituant la requête avec un sous-graphe extrait des images de la base. Nos requêtes seront partielles c'est-à-dire porteront sur une partie d'une image (connexe ou non). Nous ne considérons pas de requête constituée 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 zone d'intérêt n'est pas nécessairement le même dans l'image requête et dans l'image cible.

Nous proposons une distance entre deux sous-graphes représentant un ensemble de régions requêtes et un ensemble de régions cibles, ces ensembles pouvant être de cardinalités différentes. Cette distance prend en compte la similarité entre région requête et région cible ainsi que les positions relatives des régions des deux images. Elle permet d'ordonner les images de la base en fonction de leur similarité à la requête.

Pour être en mesure de trouver une image cible en un temps acceptable, nous proposons un algorithme de mise en correspondance inexacte de sous-graphes qui détermine une solution optimale à ce problème NP-complet, en éliminant rapidement les images et les régions trop dissemblables à la requête.

L'article est donc ainsi structuré : dans le paragraphe 2, nous présentons notre définition de la segmentation floue, ainsi que notre algorithme

pour construire une segmentation floue. Le paragraphe 3 est consacré à la phase d'indexation des images, qui consiste à coder le graphe d'adjacence des régions et les attributs des régions floues. Dans la partie 4, nous présentons notre algorithme de mise en correspondance de graphes de régions, ainsi que les dissimilarités employées pour ordonner les images résultats. Puis nous montrons des résultats sur plusieurs bases d'images dans la partie 5 avant de conclure.

2. SEGMENTATION FLOUE

Le terme de "segmentation floue" a été utilisé dans plusieurs articles, mais il n'est jamais clairement défini. Comme mentionné par Bezdek [4], les principaux moyens d'obtenir une segmentation floue, sont le seuillage flou, la classification floue, la croissance de région, ou les règles floues.

Le seuillage flou a pour but de définir des fonctions d'appartenance aux régions, basé sur un ensemble de seuils [9][31]. La plupart des méthodes de classification floue des pixels sont basées sur les c-moyennes floues (FCM) [3]. Toutefois cet algorithme ne mène pas à une segmentation proprement dite, puisqu'il classe les pixels dans des classes floues mais il ne crée pas de régions floues. Il est employé dans [5] pour réaliser une segmentation nette sur des images tomographiques cardiaques. Le nombre de classes est souvent fixé a priori. Dans les méthodes de croissance de régions, le problème est de déterminer les germes des régions, la fonction définissant l'homogénéité des régions et les fonctions d'appartenance. Moghaddamzadeh et al. [30] ont développé un algorithme complexe de croissance de régions limitées par les contours. L'algorithme comprend deux étapes, la première exécute une segmentation grossière, puis les germes sont choisis selon un double critère : fréquence des couleurs et distance aux contours. Dans la deuxième étape, qui mène à une segmentation plus fine, les germes sont des pixels les plus éloignés de toutes les régions obtenues lors de la première étape. La fonction d'appartenance tient compte de l'homogénéité en couleur et de l'éloignement aux germes. Dans [42] la croissance de région est effectuée à l'aide de règles floues impliquant des critères flous tels que l'homogénéité des régions, la taille des régions ou la valeur du gradient.

2.1. Définition de la segmentation floue

La segmentation d'image est définie comme une partition de l'image en régions, qui sont des ensembles nets (par opposition aux ensembles flous). Il y a un choix univoque pour chaque pixel d'appartenir ou non à chaque région. Mais un pixel dans « le centre » d'une région homogène, appartient certainement beaucoup plus à cette région que des pixels agglomérés à la fin d'une croissance de régions par exemple. Et un pixel près d'un contour n'est certainement pas dans la même région qu'un pixel placé de l'autre côté de ce contour. C'est pourquoi nous proposons de construire des régions qui sont des ensembles flous, avec pour chaque pixel de l'image un degré d'appartenance à chaque région.

L'algorithme que nous proposons est une méthode coopérative des approches contour et région de la segmentation [11]. Nous souhaitons extraire grossièrement les zones uniformes en couleur même si les contours en sont imprécis. Afin de profiter des avantages des deux types d'approches, nous effectuons une croissance de région, sur une image de norme de gradient. Les zones de faible gradient constituent les germes des régions. Les lignes de fort gradient seront considérées comme des barrages, difficiles à franchir. Elles pourront cependant être contournées, ce qui permet de s'affranchir du bruit impulsif.

Les degrés d'appartenance des pixels aux régions seront calculés sur un double critère de proximité spatiale et de différence de gradient avec les germes des régions. Ils sont basés sur la « distance topographique » définie dans [34].

Définition : Soit Ω un référentiel fini (ensemble de N pixels). Une segmentation floue de Ω est un ensemble de M régions floues R_j dont les supports sont inclus dans Ω et vérifiant les deux axiomes suivants. Si $\mu_j(s)$ est le degré d'appartenance du pixel s à la région R_j , alors :

1. $\forall s \in \Omega, \forall j, \mu_j(s) \in [0, 1]$
2. $\forall j, \sum_{s \in \Omega} \mu_j(s) \in]0, M[.$

Les degrés d'appartenance sont compris entre 0 et 1, ils valent 1 pour les pixels du noyau et 0 pour les pixels n'appartenant pas à la région floue. Le second axiome signifie qu'une région floue n'est ni vide, ni complète (égale à Ω). Les régions peuvent se chevaucher.

Cette définition est basée sur la définition de Ruspini d'une partition floue [39], mais sans le troisième axiome qui impose que, pour chaque pixel, la somme des degrés d'appartenance à toutes les régions soit égale à 1. Cet axiome n'est en effet pas intéressant en segmentation comme montré dans [24], car il peut créer des disparités fortes entre pixels voisins, si l'un appartient à une autre région et pas l'autre et il augmente (jusqu'à la valeur 1) les degrés d'appartenance de pixels loins de tout noyau, s'ils appartiennent à une seule région.

2.2. Algorithme de construction de la segmentation floue

La première étape consiste à appliquer un algorithme de ligne de partage des eaux [43] sur l'image des normes du gradient. Cet algorithme, appliqué sans image de marqueur, mais en prenant comme germe de bassin versant tout minimum local de la norme du gradient produit un nombre de bassins versants considérable. La deuxième étape fusionne les bassins versants en prenant comme critère l'aire ou la profondeur du bassin au moment où deux bassins entrent en contact. Chaque ensemble de bassins fusionnés donne naissance à une région floue, les degrés d'appartenance sont calculés en fonction de leur distance topographique aux germes de ces bassins. L'image de norme du gradient est obtenue par la méthode de Di Zenzo [15] avec des gradients marginaux calculés sur chaque canal couleur par le filtre de Shen-Castan [11].

On note $g(s)$ la norme du gradient du pixel s . L'image des normes du gradient est inondée dans l'ordre croissant de ces normes h ; tout pixel de niveau h est soit intégré à un bassin déjà existant s'il est connexe à ce bassin, soit constitue le germe d'un nouveau bassin. Ce germe peut être constitué de plusieurs pixels de même niveau h . Si le pixel est connexe à plusieurs bassins (il appartiendrait à la ligne de partage des eaux), il est intégré à l'un des bassins, l'aire et la profondeur du bassin sont calculées et stockées.

Pour un stockage des degrés d'appartenance sur un octet, ceux-ci prendront des valeurs entières comprises entre 0 (pour les pixels n'appartenant pas à la région) et 255 (pour les pixels du noyau). Les degrés d'appartenance au bassin B , notés μ_B sont initialisés à 255 pour les pixels appartenant aux germes, les autres prennent la valeur 0. Les pixels du germe sont stockés dans la file Q_B .

Initialisation des degrés d'appartenance lors de l'inondation

Pour chaque bassin B
 pour chaque pixel $s \in B$
 si $s \in \text{germe}(B)$ $\mu_B(s) \leftarrow 255$, mettre s dans Q_B
 sinon $\mu_B(s) \leftarrow 0$
 fin pour
 Fin pour

Après la fin de l'inondation, les bassins versants sont fusionnés en fonction de leurs aire et profondeur [1] : un bassin trop petit ou trop peu profond est absorbé par son voisin. On note $Absorb(B)$ le bassin absorbant B . De plus une pénalité sur les degrés d'appartenance est appliquée sur le bassin absorbé, cette pénalité est égale à la différence entre les niveaux des fonds des deux bassins.

Fusion des bassins

Pour chaque pixel $s \in \text{germe}(B)$
 $h_B \leftarrow g(s)$
 tant qu'il existe A tel que $A = Absorb(B)$
 $A \leftarrow Absorb(B)$, $B \leftarrow A$
 $h_A \leftarrow g(t)$, $t \in \text{germe}(A)$
 $\mu_B(s) \leftarrow 255 - (h_B - h_A)$
 Fin pour

Une région floue est construite à partir d'un ensemble de bassins fusionnés. Les degrés d'appartenance sont calculés à partir des degrés d'appartenance des germes des bassins. Ils sont maximaux (255) pour les pixels appartenant aux germes des bassins, éventuellement diminués de la différence de hauteur entre les fonds des bassins. Ils diminuent ensuite au fur et à mesure que l'on s'éloigne des germes : ils perdent 1 pour chaque éloignement spatial de 1 pixel (pris en 4-connexité) et ils perdent une valeur proportionnelle à la différence entre les normes de gradient, jusqu'à atteindre la valeur 0.

<p>Calcul des degrés d'appartenance à la région R</p> <p>Pour chaque bassin B de R</p> <p> tant que Q_B est non vide</p> <p> extraire s de Q_B</p> <p> pour chaque pixel v voisin de s</p> <p> $\mu = \mu_B(s) - (\lambda \cdot g(v) - g(s) + I)$</p> <p> si $\mu > \mu_B(v)$ alors $\mu_B(v) \leftarrow \mu$</p> <p> mettre v dans Q_B</p> <p> fin pour</p> <p> fin tant que</p> <p>Fin pour</p>
--

Ainsi les pixels appartenant à des zones homogènes en couleur, donc de faible gradient auront de forts degrés d'appartenance à la région floue correspondante. Ces degrés diminuent lentement au fur et à mesure de l'éloignement du noyau de la région et diminuent fortement à la rencontre d'un contour, zone de fort gradient. Le degré d'appartenance d'un pixel est inversement proportionnel à la distance topographique telle que définie dans [34], longueur du plus court chemin reliant le pixel au noyau, le long de la surface que constitue la norme du gradient dans l'espace 3D. Le bruit impulsionnel est ainsi contourné. Les régions floues se chevauchent plus ou moins (cf. Fig. 12).

L'algorithme fait appel à trois paramètres : λ règle le degré de flou et donc l'extension des régions, il peut varier de 1 à 3, nous l'avons fixé à 2 dans tous nos tests. Le seuil de profondeur qui intervient dans la fusion a été fixé à 3 dans tous nos tests. Le dernier paramètre est le seuil sur l'aire des bassins à fusionner, c'est de lui que dépend le nombre de régions. Nous le faisons varier par dichotomie (en le doublant ou le divisant par deux) à partir d'une valeur initiale, pour atteindre un nombre de régions indiqué par l'utilisateur. Celui-ci définit un intervalle du nombre de régions qu'il souhaite pour les images de la base et l'algorithme ajuste le seuil sur l'aire afin de rester dans l'intervalle souhaité par l'utilisateur. Nous avons fixé un nombre de régions compris entre 5 et 25 pour la base généraliste utilisée ci-dessous.



Une image de notre base de données généraliste

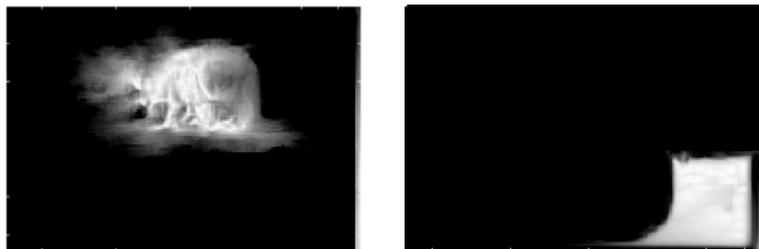


Fig. 1 – Deux régions floues obtenues à partir de l'image du haut ; à gauche, région correspondant à l'ours
en bas, région correspondant à une partie de l'eau
(plus c'est clair, plus le degré d'appartenance est élevé)

La figure 1 présente un exemple de régions floues obtenues par l'algorithme. 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. On peut obtenir une segmentation nette à partir de cet algorithme en affectant chaque pixel à la région pour laquelle il a le plus fort degré d'appartenance. On n'utilise cette « defuzzification » que pour afficher simultanément toutes les régions floues (cf. Fig. 8).

3. SIGNATURE DES IMAGES

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.

Rosenfeld a étendu la définition des attributs géométriques (aire, périmètre, etc.) aux ensemble flous [38]. Le principe est de pondérer la contribution de chaque pixel s par son degré d'appartenance $\mu(s)$ à la région R . Nous utiliserons le centre de gravité G de la région R , défini par ses coordonnées :

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

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 [16]. La probabilité pour tout réel r est définie par :

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

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 la distribution.

Nous utilisons dans la suite la distribution colorimétrique en k classes. Ces classes de couleur sont obtenues par l'algorithme des k -means, calculé sur tous les pixels de la base, à partir de leurs composantes RVB. Le nombre de classes est très variable selon les systèmes : si le découpage est statique, on trouve par exemple 166 dans [41], 25 dans [35], et il s'agit de découpages réguliers. Des tests menés dans [18] [19] avec un nombre de classes variant de 8 à 400 ont montré que 25 est un bon compromis entre compacité de la signature et efficacité et le choix des 25 centres de classes est ici adapté à la base. Une palette adaptée à l'image [17] semble être aussi une solution à envisager. 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.

Il existe différents moyens de représenter les relations spatiales entre régions. Les régions elles-mêmes peuvent être représentées à l'aide d'un point (en général leur centre de gravité), d'un rectangle (englobant par exemple), etc. En indexation d'images, une des premières représentations des relations spatiales entre objets a employé des 2D Strings [8], qui sont obtenues par projection des objets sur les axes de coordonnées. Malki et al. [27] emploient les relations d'Allen [1] pour définir des relations topologiques et d'orientation (telles que chevauchement, recouvrement, disjonction) entre objets. Matsakis et al. [28] étendent la notion d'histogrammes d'angles [25] et stockent dans des histogrammes de force les informations de position directionnelle relative de deux objets. Des attributs flous de positions relatives ont été proposés dans [6] pour des ensembles nets et dans [25] pour des ensembles flous. D'après nos tests, ces attributs flous ont l'inconvénient d'être très longs à calculer sans amélioration notable des résultats. Le problème de représenter les relations spatiales entre objets fait donc l'objet de nombreux travaux récents, et la comparaison entre ces différentes méthodes dans le cadre de la recherche d'images par le contenu reste à faire.

Dans cet article, nous ne nous focalisons pas sur le choix de la représentation la plus appropriée, les méthodes citées ci-dessous sont souvent trop précises pour rechercher un ensemble de régions ayant des formes très variables d'une image à l'autre, de plus elles ne s'adaptent pas nécessairement simplement aux régions floues. Nous proposons un moyen de mettre en correspondance des ensembles de régions situées grosso modo dans la même disposition d'une image à l'autre. Par exemple dans une voiture vue de profil, les roues sont situées sous la carrosserie, les vitres en haut et le tout se trouve en général sur une zone de goudron.

Des tests dans lesquels nous avons affecté des valeurs précises aux arêtes telles que les relations floues [6] ou plus simplement les distances entre centres de gravité nous ont montré la non-pertinence d'une telle approche compte-tenu de la grande variabilité des objets recherchés. C'est pourquoi, la position relative des régions est simplement prise en compte au travers des positions relatives des centres de gravité, réduites aux deux relations binaires : « au-dessus de » et « à gauche de ».

La signature de l'image est donc constituée de deux parties, d'une part des informations sur les régions : aire, distribution couleur, coordonnées du centre de gravité et d'autre part la matrice d'adjacence des régions :

pour des régions floues, l'adjacence vaut 1 si les deux régions ont au moins un pixel en commun, 0 sinon.

4. 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. Les nœuds du graphe correspondent aux régions et les arêtes aux adjacences entre régions. Les régions sont valuées (par leurs attributs de couleur, de texture, etc.), les arêtes sont valuées par les informations binaires « au-dessus de » et « à gauche de ».

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 (cf. Fig. 4).

L'algorithme de mise en correspondance que nous proposons 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 identiques en ce qui concerne les arêtes (même valeur d'adjacence et de position spatiale relative).

4.1. Systèmes de recherche d'images basés sur les régions

Les systèmes d'indexation d'images employant des régions ne prennent en général pas en compte la composition de l'image [7] [22] [23] 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 ré-

gions des deux images. Souvent la topologie n'est représentée que par la position absolue [17] ou relative des centres de gravité des régions. Dans le système Picasso [14], 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 [23], 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 [44], 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. Prasad et al. [35] emploient une table de hashage pour accéder rapidement aux images, le codage de la signature est alors très rigide : pour conserver l'information spatiale, l'image segmentée est découpée en 9 blocs de même taille et seul le numéro de la région dominante dans chaque bloc est stocké.

L'appariement de graphes a été depuis longtemps utilisé en reconnaissance des formes. La relaxation [37] permet de réduire un étiquetage (discret, flou ou probabiliste) par la prise en compte des étiquettes des nœuds voisins. Les graphes ont aussi été employés pour la recherche d'image. Hong et Huang [21] 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 [36] 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. Medasani et Krishnapuram [29] effectuent un appariement inexact de graphes valués flous, dans lesquels les régions sont affectées d'étiquettes symboliques. Les positions relatives des régions sont prises en compte au travers des relations floues de Bloch [6]. Pour limiter le temps de calcul

des appariements, les graphes sont préalablement regroupés en « clusters » de graphes, représentés par un prototype, et seuls les graphes appartenant au « cluster » le plus proche de la requête sont comparés. Perchant et Bloch [33] ont défini un formalisme général du morphisme flou entre graphes. Notre algorithme entre dans ce cadre puisqu'il définit une fonction sur les couples de nœuds appariés (une distance entre régions, présentée au § 4.3) et une fonction sur les couples d'arêtes appariées, prenant des valeurs binaires selon que les arêtes indiquent les mêmes positions relatives dans l'image requête et dans l'image cible.

La mise en correspondance peut se faire avec un nombre de régions différentes entre la requête et la cible. Dans SIMPLicity [44], 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 [41], 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.

Dans notre système, les positions relatives des régions sont prises en compte dans la mise en correspondance des graphes de régions pour réduire la combinatoire. En effet nous cherchons à résoudre un problème NP-complet de mise en correspondance de graphes valués, avec un nombre de nœuds qui n'est pas nécessairement égal dans les deux graphes. De plus la solution doit être trouvée en un temps acceptable pour un utilisateur effectuant une requête sur une base.

Cordella et al ont présenté un algorithme de mise en correspondance inexacte entre graphes valués [12] : 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.

4.2. Mise en correspondance de graphes appliquée à la recherche d'image par arbre de recherche

De même que Cordella et al. [12], nous utilisons un arbre de recherche (Fig. 2) et nous proposons une solution pour la construction rapide de cet arbre adapté au problème de mise en correspondance d'ensembles de

régions. Notre but est d'obtenir la solution optimale pour une fonction de similarité donnée (voir § 4.3).

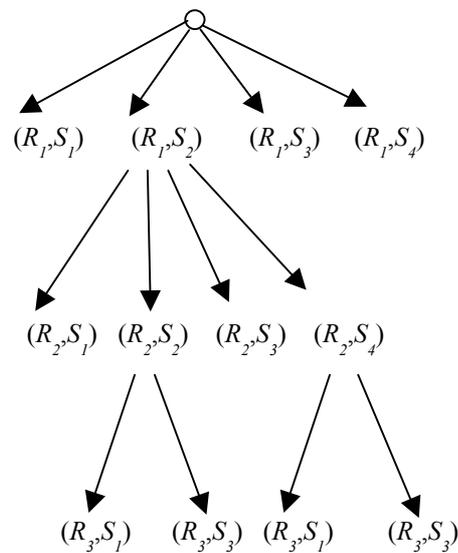


Fig. 2 – *Arbre de recherche : chaque nœud correspond à un appariement de deux régions (R_i : région requête, S_j : région cible), chaque flèche correspond à une compatibilité topologique*

Pour chaque image de la base (image cible) les distances entre toutes les paires région-requête, région-cible sont calculées.

L'arbre est constitué de nœuds représentant chacun 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 Fig. 2, 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.

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 4.3. 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. Voici ce que nous proposons :

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 plus différente de toutes les régions cibles) 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. 2 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 éliminée.
- 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.

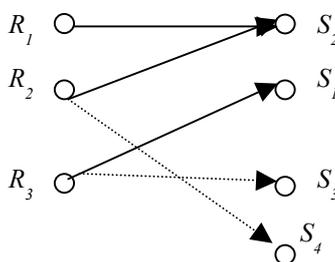
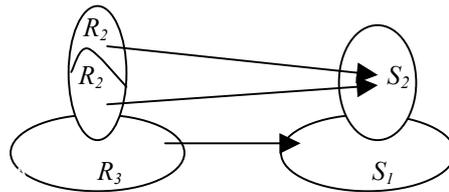


Fig. 3 – Mise en correspondance initiale, correspondant à un chemin de la Fig. 2 (flèches continues), complété par la deuxième étape (flèches en pointillé).

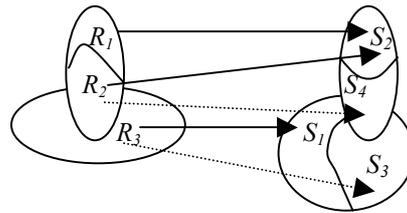
Chaque chemin de la racine à une feuille du dernier niveau représente un appariement de sous-graphes acceptable. Par exemple dans la Fig. 2, $\{(R_1, S_2), (R_2, S_2), (R_3, S_1)\}$ est l'un des quatre appariements de graphes possibles. On remarque qu'une région de l'image cible peut être appariée à plusieurs régions requête, mais pas l'inverse (Fig. 2). On peut donc gérer une sous-segmentation de l'image cible par rapport à l'image requête (Fig. 4a), mais pas une sur-segmentation (Fig. 4b).

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. 4b 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 de la Fig. 4, 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 .



a. Exemple d'une image cible sous-segmentée par rapport à l'image requête



b. Exemple d'une image cible sur-segmentée correspondant à l'appariement complet de la Fig. 3
(flèches continues : première étape, flèches pointillées : deuxième étape)

Fig. 4 – Deux exemples d'appariements de graphes

4.3. MESURES DE DISSIMILARITE ENTRE REGIONS ET ENTRE SOUS-GRAPHES

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 n'ont aucune classe couleur en commun.

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.

Considérant que toutes les régions ont la même importance, nous avons choisi comme mesure de dissimilarité la simple moyenne des dissimilarités des régions appariées.

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-graphes comprenant n ($n \geq m$) appariements de régions est :

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

La pondération par des valeurs prenant en compte plus finement la compatibilité des positions relatives des régions n'a pas amélioré les per-

formances, ceci à cause de la variabilité des objets recherchés. D'autre part une pondération par exemple par l'aire de la région ne nous paraît pas pertinente car toute région, même petite a son importance dans la reconnaissance et il ne faudrait pas perdre l'intérêt de la segmentation en diminuant la contribution des petites régions.

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. RESULTATS

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 Columbia [45] qui contient 100 objets différents vus chacun sous 36 angles. Le problème est de retrouver des images contenant un objet quelle que soit sa position dans l'image. La Fig. 5 montre un résultat avec une requête faite de trois régions floues constituant la tasse. Toutes les images retrouvées dans les 16 premières contiennent cette tasse malgré des changements d'orientations et de petites variations d'échelle.

Les autres résultats ont été obtenus avec une base de données généraliste de 1200 images automatiquement segmentées (cf. Fig. 1).

Dans la Fig. 6, 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. 1). La Fig. 6 présente les images ordonnées. On obtient 11 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

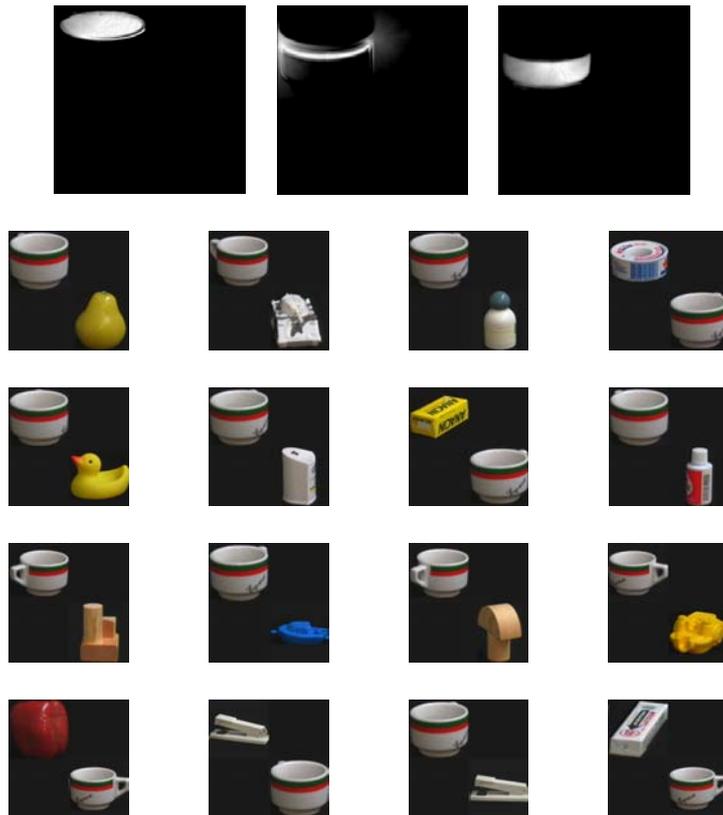


Fig. 5 – La requête est constituée de trois régions (première ligne) formant la tasse (base Columbia). Les images sont classées par dissimilarité croissante de haut en bas et de gauche à droite.

distribution colorimétrique est prise en compte et qu'il n'y a pas de bouclage de pertinence, d'où la confusion entre l'eau et le ciel notamment.

Fig. 7 sont représentées les régions floues pour les deux meilleures images. On note que ces régions 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 très variables 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 »).

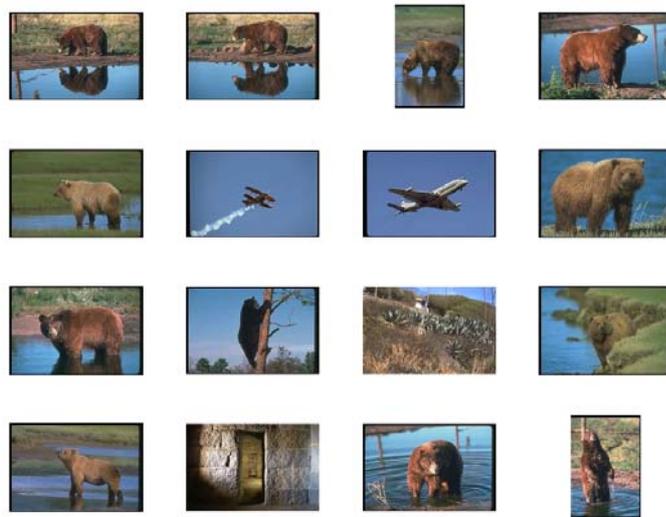


Fig. 6 – 16 premières images à partir d'une requête constituée des deux régions de la Fig. 1 (l'ours au-dessus de l'eau). 11 images correctes.

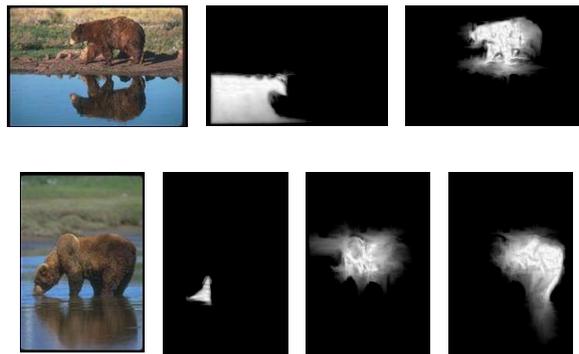


Fig. 7 – Les deux premières images retrouvées et pour chacun d'elles, les régions appariées aux deux régions requête.

voit sur cet exemple l'intérêt de la segmentation floue, qui permet une grande souplesse dans la forme des régions.

Un autre exemple présenté Fig. 8 montre que l'on peut retrouver les voitures rouges de la base à partir d'une requête constituée d'une partie de la carrosserie et d'une roue. Parmi les 25 images les plus proches, 24 contiennent une voiture rouge, prise sous différents angles. La Fig. 9 présente la courbe de précision/rappel pour cette requête.

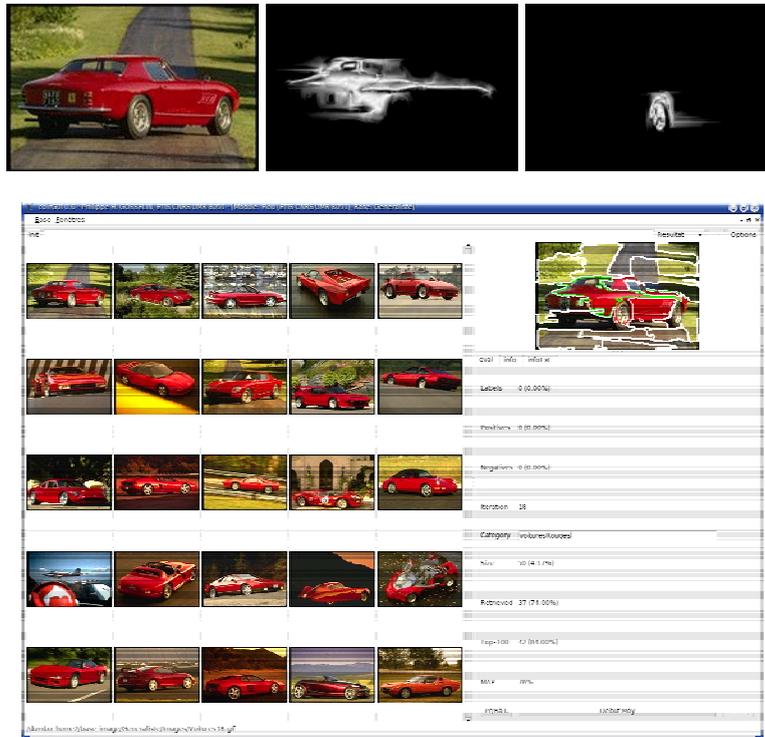


Fig. 8 – Recherche de voitures rouges à partir d'une requête constituée des 2 régions présentées en haut. Parmi les 25 images les plus proches de la requête, 24 contiennent une voiture rouge.

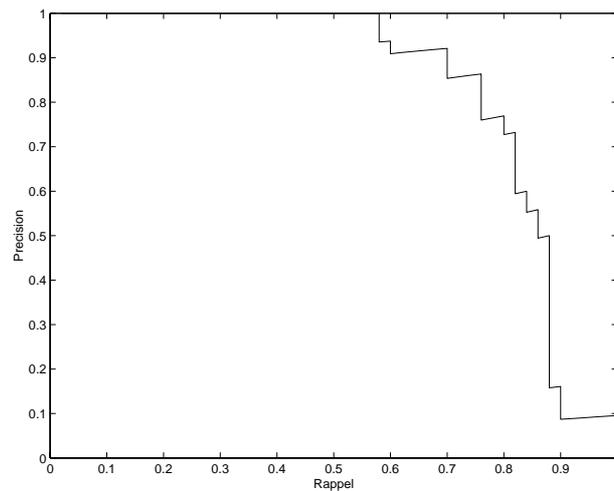


Fig. 9 – Courbe de précision/rappel pour la catégorie « voiture rouge ». La partie droite de la courbe, après le décrochement (rappel de 0,85) correspond aux images écartées lors de la construction de l'arbre et pour lesquelles aucune distance n'a été calculée.

Puis nous avons comparé ces résultats avec une recherche globale par un classifieur SVM avec un noyau gaussien et une distance du χ_1 [13]. La signature image pour le SVM est la distribution colorimétrique dans les mêmes 25 classes couleur que pour les régions et le classifieur démarre sur une seule image de la catégorie afin de fonctionner dans les mêmes conditions que le système à base de régions. Les courbes de la Fig. 10 présentent des résultats de la catégorie « voiture rouge », moyennés sur 50 recherches effectuées avec une, deux ou trois régions couvrant la voiture rouge. On peut conclure sur cet exemple que si on dispose d'un seul exemple de la catégorie recherchée, notre système basé sur des régions fonctionne mieux qu'un SVM. Ceci s'explique par le fait que seules les distributions colorimétriques associées à l'objet d'intérêt sont prises en compte dans le calcul de la dissimilarité et non la distribution colorimétrique de l'ensemble de l'image comme dans la recherche globale par SVM.

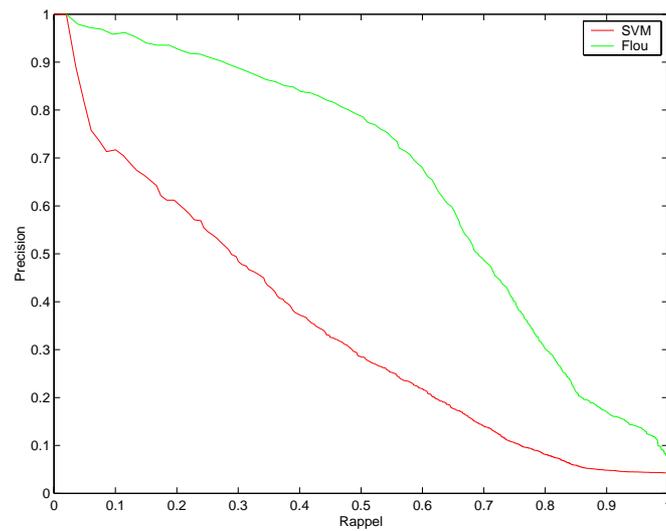


Fig. 10 – Comparaison requête partielle / classifieur SVM sur la catégorie voitures rouges. Moyenne sur 50 recherches

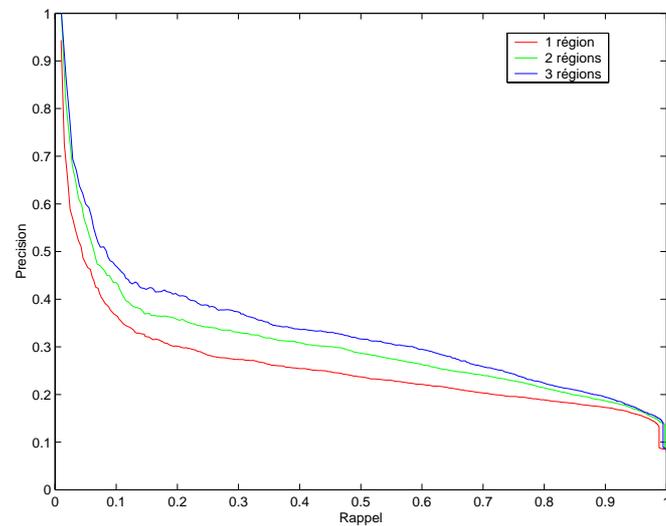


Fig. 11 – Comparaison de requête sur une, deux et trois régions. Moyenne sur 50 requêtes prises dans des images de la catégorie éléphants

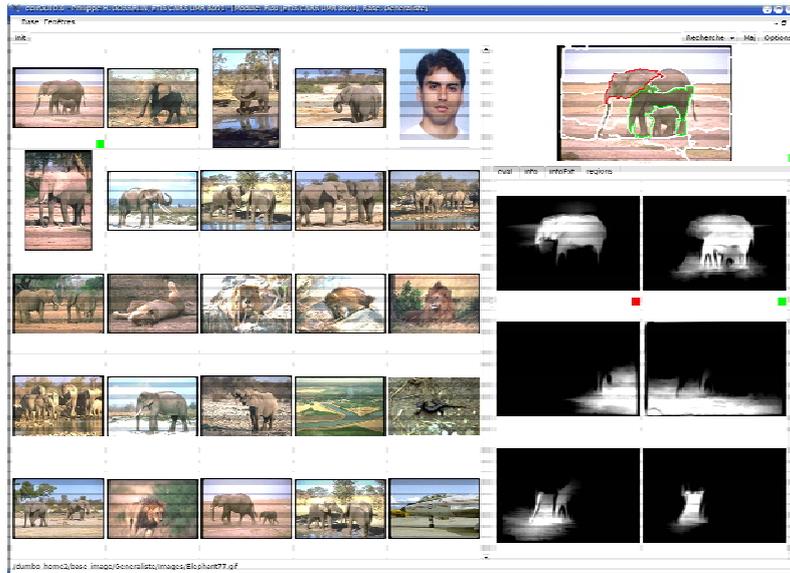


Fig. 12 – Résultat à partir d'une requête constituée de deux régions floues couvrant l'éléphant. A droite en haut, l'image requête avec les contours de l'image segmentée « défuzzifiée », au dessous les deux régions requête. A gauche, les images résultats classées.

Nous avons ensuite étudié le lien entre le nombre de régions requête et les résultats. Nous avons comparé la recherche d'images contenant des éléphants à partir d'une, de deux et de trois régions et nous avons moyenné sur 50 recherches de chaque type (sur différentes images de la catégorie éléphant). Un exemple de requête sur deux régions est présenté Fig. 12. Ce résultat (Fig. 11) ainsi que d'autres tests similaires montrent qu'on a intérêt à couvrir au mieux l'objet recherché, et qu'une redondance d'information (due au recouvrement des régions) est préférable à une information partielle contenue dans une seule région ne couvrant pas complètement l'objet.

Cet algorithme réduit considérablement la combinatoire du nombre d'appariements à examiner. Si le nombre de régions formant la requête est égal à m et le nombre de régions dans l'image cible est égal à p , le nombre d'appariements possibles de régions est $p \times m$ et le nombre maximum de chemins dans l'arbre est p^m . 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 $m/2$ et $m \times 2$. Comme le nombre de régions requête est faible (inférieur à cinq), le temps de traitement est compatible avec une utilisation en temps réel.

6. CONCLUSION ET PERSPECTIVES

Nous avons d'abord proposé un algorithme de segmentation floue, le premier du genre à notre connaissance. Il effectue une croissance de régions initialisée par une image des normes de gradient. Il intègre donc les avantages des approches région et contour de la segmentation d'image : obtention de régions fermées et segmentation complète de l'image avec contrainte sur les bords des régions par les fortes valeurs de gradients.

L'intérêt de cette méthode n'est pas dans la segmentation, puisqu'on n'obtient pas une partition nette de l'image, mais dans les régions floues elles-mêmes. Ces ensembles flous sont en effet très appréciables, tant en reconnaissance des formes qu'en indexation d'images, puisque dans ces deux domaines, on n'a pas besoin d'une segmentation précise, mais bien de pouvoir calculer des attributs sur des zones de l'image, même si les contours en sont imprécis.

Cet algorithme s'applique aussi bien à des images couleur que monochromes. Il ne nécessite aucune image de germes de régions, puisque celles-ci sont créées dynamiquement. Les quelques seuils intervenant dans l'algorithme peuvent être fixés a priori. Le nombre approximatif du nombre de régions souhaité, et qui fixera la finesse de la segmentation est laissé au choix de l'utilisateur. Dans le cadre de la segmentation automatique d'une base entière, l'utilisateur indique une fourchette pour le nombre de régions souhaité.

Nous avons ensuite proposé une méthode de recherche d'image à partir d'une requête partielle, constituée de plusieurs régions d'une image. L'algorithme développé effectue une mise en correspondance de sous-

graphes et a pour principal avantage de réduire la combinatoire et donc d'être 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.

Nous avons montré sur des exemples que notre système permet de retrouver des images contenant un type d'objet ou d'animal avec une grande variabilité de position (vue de face, de profil, etc.) et d'échelle. La requête est de type partiel (quelques régions d'une image pointées par l'utilisateur), le système renvoie des images avec les régions mises en correspondance avec les régions requête. Mais le système peut aussi servir à une recherche globale, puisqu'il donne une mesure de similarité globale de toute image avec la requête (qui reste partielle), permettant d'extraire d'une base les images contenant un objet (ou animal) particulier dans n'importe quel contexte. Nous avons montré sur un exemple que notre système est dans ce cas tout à fait concurrentiel avec les systèmes les plus performants actuels qui emploient des recherches globales, l'atout de notre système étant de ne prendre en compte que l'objet dans la recherche.

Nous avons présenté des résultats avec un ensemble d'attributs simples (la distribution colorimétrique et la position des centres de gravité) et une distance simple. La palette couleur adaptative proposée dans [17] avec la distance associée nous paraît une alternative prometteuse. 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. Enfin un bouclage de pertinence permettra de renforcer les ensembles de région pertinentes et de mieux éliminer les autres, offrant ainsi au système de meilleures capacités à la généralisation [20].

Références

- [1] J-F. Allen, Maintaining knowledge about temporal intervals, *Comm. of the ACM*, 11 (26), 832-843, 1983
- [2] M.C. De Andrade, G. Bertrand, A.A. Araujo, Segmentation of microscopic images by flooding simulation: A catchment basins merging algorithm. *Proc. of the IST/SPIE Symp. On Electronic Imaging, Science & Technology*, San Jose-CA, USA, 1997

- [3] J.C. Bezdek, Pattern recognition with fuzzy objective function algorithms, *Plenum Press*, New York, 1981
- [4] J.C. Bezdek, J. Keller, Krisnapuram, S. K. Pal, Fuzzy models and algorithms for pattern recognition and image processing, ed. *Kluwer*, 2000
- [5] N. Boujemaa, G. Stamon, J. Lemoine, Fuzzy iterative image segmentation with recursive merging. *SPIE 1818, Visual communication and image processing*, 1271-1281, 1992.
- [6] I. Bloch, Fuzzy relative position between objects in image processing : a morphological approach, *IEEE Int. Conf. on Image Processing*, Lausanne, 987-990, 1996
- [7] 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
- [8] S. K. Chang, Q. Y. Shi, C. W. Yan, Iconic indexing by 2-D strings, *IEEE Trans. on PAMI*, 9 (3), 413-428, 1987
- [9] H.D. Cheng, Y-H. Chen, Fuzzy partition of two-dimensional histogram and its application to thresholding. *Pattern Recognition* 32, 825-843, 1999
- [10] Y. Chen, J. Z. Wang, Image categorization by learning and reasoning with regions, *J. of Machine learning* 5, 93-939, 2004
- [11] J-P. Cocquerez, S. Philipp coordinateurs Analyse d'images : filtrage et segmentation, ed *Masson*, sept. 1995
- [12] L. P. Cordella, P. Foggia, C. Sansone, M. Vento, Subgraph Transformation for the inexact Matching of Attributed Relational Graphs, *Computing*, 12, 43-52, 1998.
- [13] M Cord, S Philipp-Foliguet, P-H. Gosselin, J Fournier, Interactive exploration to image retrieval, *Journal of Applied Signal Processing*, vol 2005, issue 14, Special Issue on Advances in Intelligent Vision Systems: Methods and Applications—Part II, 2005
- [14] A. Del Bimbo, M. Mugnaini, P. Pala, F. Turco, Visual Querying by color perceptive regions, *Pattern Recognition*, 31 (9), 1241-1253, 1998
- [15] S. Di Zenzo, A note on the gradient of a multi-image, *CVGIP: Graphics and image processing*, 33, 116-125, 1986
- [16] D. Dubois, M.C. Jaulent, A general approach to parameter evaluation in fuzzy digital pictures, *Pattern Recognition Letters* 6, 251-259, 1987
- [17] J. Fauqueur, N. Boujemaa, Recherche d'images par régions d'intérêt : segmentation grossière rapide et description couleur fine, *TSI*, numéro spécial Indexation de Bases d'Images fixes ou animées, n°22 (9), 1107-1138, 2003
- [18] J. Fournier, M. Cord, S. Philipp-Foliguet, RETIN : A content-based image indexing and retrieval system, *Pattern Analysis and Applications*, special issue on image indexation, n°4, 153-173, 2001

- [19] 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>
- [20] J. Gony, Recherche interactive d'image dans une base de données images à partir d'une requête partielle, stage de Master IAD, ETIS, 2005
- [21] 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
- [22] F. Jing, M. Li, H-J. Zhang, B. Zhang, An efficient and effective region-based image retrieval framework, *IEEE Trans. on PAMI*, 13 (5), 699-709, 2004
- [23] I. Kompatsiris, E. Triantafillou, G. Strintzis, Region-based color image indexing and retrieval, *IEEE ICIP*, Thessaloniki, Greece, 658-661, 2001
- [24] R. Krishnapuram, J. M. Keller, A possibilistic approach to clustering, *IEEE Trans. on Fuzzy Systems*, 1 (2), 98-110, 1993
- [25] R. Krishnapuram, J. M. Keller, Y. Ma, Quantitative analysis of properties and spatial relations of fuzzy image regions, *IEEE Trans. on Fuzzy Systems*, 1 (3), 222-233, 1993
- [26] W. Y. Ma, B. S. Manjunath, NeTra : a toolbox for navigating large image databases, *ACM Multimedia Systems*, 7 (3), 184-198, 1999
- [27] J. Malki, E. Zahzah, L. Mascarilla, Indexation et recherche d'image fondées sur les relations spatiales entre objets, *Traitement du signal*, 19 (4), 235-250, 2002
- [28] P. Matsakis, J.M. Keller, O. Sjahputera, J. Marjamaa, The use of force histograms for affine-invariant relative position description, *IEEE Trans on PAMI*, 26 (1), 1-18, 2004
- [29] S. Medasani, R. Krishnapuram, A fuzzy approach to content-based image retrieval, *IEEE Int. Fuzzy systems Conf. Proc*, Seoul, 1251-1260, 1999
- [30] A. Moghaddamzadeh, N. Bourbakis, A fuzzy region growing approach for segmentation of color images. *Pattern Recognition*, 30 (6), 867-881, 1997
- [31] N. R. Pak, S. K. Pal, A review on image segmentation techniques. *Pattern Recognition* 26 (9), 1277-1294, 1993
- [32] 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
- [33] A. Perchant, I. Bloch, Fuzzy morphisms between graphs, *Fuzzy Sets and Systems*, 128 (2), 2002
- [34] S. Philipp-Foliguet, M. B. Vieira, A. de A. Araújo, Segmentation into fuzzy regions using topographic distance, *14th SIBGRAPI*, 282-288, Florianopolis, Brazil, 2001

- [35] B. G. Prasad, K. K. Biswas, S. K. Gupta, Region-based image retrieval using integrated color, shape and location index, *CVIU*, Special issue on colour for image indexing and retrieval, vol 94, 193-233, 2004
- [36] A. Robles-Kelly, E. R. Hancock, Graph matching using adjacency matrix Markov chains, *Proc. 3rd Int. Workshop EMMCVPR*, Sophia-Antipolis, France, 2001
- [37] A. Rosenfeld, A., Hummel, R.A., and Zucker, S.W., Scene Labeling by Relaxation Operations, *IEEE Trans on SMC*, 6(6), 420-433, 1976
- [38] A. Rosenfeld, The fuzzy geometry of image subsets. *Pattern Recognition Letters* 2, 311-317, 1984.
- [39] E.H. Ruspini, A new approach to clustering. *Information and Control*, 15 (1), 22-32, 1969.
- [40] 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
- [41] J. R. Smith, S-F. Chang, VisualSEEK : a fully automated content-based image query system, *ACM Multimedia Conf.*, Boston, USA, 87-98, 1996
- [42] A. Steudel, M. Glesner, Fuzzy segmented image coding using orthonormal bases and derivative chain coding. *Pattern Recognition* 32, 1827-1841, 1999
- [43] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulation, *IEEE Trans on PAMI*, 13 (6), 563-598, 1991.
- [44] J. Z. Wang, J. Li, G. Wiederhold, SIMPLIcity : semantics-sensitive integrated matching for picture libraries, *IEEE Trans. on PAMI*, 23 (9), 1-17, 2001
- [45] <http://www.cs.columbia.edu/CAVE/research/softlib/coil-100.html>