République Tunisienne
Ministère de l'Enseignement Supérieur

Université de Sfax
Ecole Nationale d'Ingénieurs de Sfax
Département d'Informatique et de Mathématiques Appliquées

Cycle de Formation d'Ingénieurs
dans la discipline
Génie Informatique

Projet de fin d'études

N° d'ordre: 2006–GI03

MÉMOIRE

présenté à

l'École Nationale d'Ingénieurs de Sfax
(Département d'Informatique et de Mathématiques Appliquées)

en vue de l'obtention

du Diplôme National d'Ingénieur en Génie Informatique

par

Ali KARRAY

Développement d'une méthode d'indexation spatiale pour la recherche d'images par le contenu

soutenu le 26 juillet 2006, devant la commission d'examen:

Mr.	***Adel ALIMI***	Président
Mr.	***Jean Marc OGIER***	Encadreur
Mr.	***Slim KANOUN***	Encadreur
Melle.	***Wafa BOUSSELLAA***	Membre

بسم الله الرحمان الرحيم

A mon père et A ma mère

Auxquels je dois ce que je suis. Que dieu vous protège.

A mes deux sœurs

Qu’ils trouvent dans ce mémoire l’expression de mes remerciements les plus sincères

Remerciements

C'est avec un grand plaisir que je réserve ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Mes remerciements s'adressent à Monsieur le président ***Mohamed Adel ALIMI*** *et Mademoiselle* ***Wafa BOUSELLAA****, membre du jury, pour l'honneur qu'ils m'ont accordé en acceptant de juger mon travail.*

Je tiens aussi à remercier vivement ***Mr. Slim KANOUN****, qui n'a cessé de me prodiguer ses conseils et ses suggestions pertinentes.*

Je remercie également ***Mr.Jean-Marc OGIER*** *pour l''accueil chaleureux qu'il m'a réservé dans l e laboratoire L3i et pour sa contribution décisive dans l'élaboration de ce travail.*

A tous mes amis, en leur espérant bonne continuation dans leur travaux.

Table de matières

Introduction générale

De nos jours, les images occupent une place prépondérante au sein de la société. L'apparition des dispositifs d'acquisition, des capacités de stockage (mémoires de masse), des systèmes de transmission et de diffusion (réseaux Internet, réseaux spécialisés à haut débit) nécessite un besoin croissant en traitement de l'image.

Des bases de données ont alors été construite afin de stocker toutes ces images et au fil des années elles ont gagné en importance occupant parfois jusqu'à plusieurs téraoctets de mémoire et ayant des natures très diverses. Ainsi, le volume de ces bases d'images étant très important, le problème de la recherche et de la consultation devient très difficile. Il est donc indispensable de développer des outils permettant de sélectionner, par le contenu, les images les plus pertinentes. Le problème qui se pose alors, est de trouver un alphabet (appelé index) relatif à l'information visuelle présentée dans l'image, afin de le répertorier dans un "dictionnaire", les index ainsi définis décrivent les modèles correspondants.

Traditionnellement, des attributs textuels de l'image tels que le nom de fichier, la légende et les mots clefs sont utilisés pour indexer les bases d'images. La problématique de la recherche d'images est déplacée vers celle de la recherche des mots, que l'on sait résoudre. Ce type d'approche est parfois efficace, mais s'avère limité. Le problème repose sur le fait que l'information visuelle est complexe, celle-ci nécessite l'intervention manuelle d'un opérateur (souvent d'un expert) pour une description demeurant subjective et approximative. Dans ce cas, la limitation est non seulement liée au nombre d'images (qui ne doit pas être excessivement grand), mais aussi à la nature du contenu informatif de celles-ci qui doit être parfaitement déterminé. Un autre inconvénient subsiste en ce qui concerne la barrière linguistique alors que les images ont une signification plus "universelle". Enfin, il existe des applications où le volet d'informations à gérer dépasse le raisonnable pour une indexation textuelle.

Afin de surmonter ces problèmes, les recherches récentes sur les moteurs de recherche d'images mettent l'accent sur la recherche d'images basées sur le contenu (CBIR) qui utilise les caractéristiques de bas niveau de l'image telles que couleur, texture et forme. Quelques prototypes académiques et commerciaux ont été développés récemment pour permettre la recherche d'images par le contenu à travers les bases de données, nous pouvons en citer

Virage et Visual seek.

On distingue deux approches utilisées dans l'indexation par le contenu: l'approche globale et l'approche spatiale. La première approche considère l'image dans son ensemble et la caractérise en utilisant des statistiques calculées sur l'image entière.

L'idée de la deuxième approche est de considérer l'image comme un ensemble d'objets et non plus comme une entité unique. La représentation des données est décrite par un ensemble de chaînes, d'arbres ou de graphes. Le problème d'indexation est alors vu comme une mise en correspondance entre les deux structures.

Le présent travail s'inscrit dans le cadre de la collaboration entre les deux laboratoires REGIM (Groupe de REcherche sur les Machines Intelligentes) de l'Ecole Nationale d'Ingénieurs de Sfax et le laboratoire L3i (Informatique, Image, Interaction) de L'Université de La Rochelle. L'objectif est la mise en place d'une méthode d'indexation spatiale d'images vu qu'il n'existe pas dans ces deux laboratoires une méthode d'indexation de ce type.

Dans le premier chapitre, un état de l'art sur l'indexation de l'image par le contenu est présenté.
Le deuxième chapitre établit quelques méthodes d'appariement de graphes qui existent dans la littérature. Nous proposerons par la suite un algorithme d'apparient de graphes.
Dans le troisième chapitre, il s'agit de présenter notre application développée pour la recherche d'images synthétiques par le contenu ainsi que notre application de recherche d'images de lettrines par le contenu.

Chapitre I

Approches d'indexation d'images

1. Introduction

Durant les dix dernières années, la recherche d'images par le contenu visuel (CBIR - Content-based image retrieval) est l'un des domaines de recherche les plus actifs dans le secteur de la vision par ordinateur. L'accessibilité à des quantités importantes de documents visuels et multimédias ainsi que le développement de l'Internet font naître le besoin de méthodes d'accès aux données qui offrent plus que des recherches simples par texte. Un grand nombre d'utilitaires a été développé pour formuler et exécuter des requêtes basées sur le contenu des documents, et pour faciliter la navigation dans de grandes collections d'images. Malgré cela, jusqu'à maintenant il n'y a pas encore eu de grandes avancées au niveau de la recherche dans les bases d'images. Un des problèmes importants est la perte de l'information sémantique dans les caractéristiques extraites automatiquement. Les premières approches développées en indexation décrivaient les images dans leur ensemble. Des statistiques simples sont alors calculées pour représenter les images. Généralement, les informations prises en compte sont la couleur, la texture et/ou la forme. Malheureusement, ces techniques décrivent les images globalement. Il paraît difficile avec ce type d'approche d'obtenir une description plus fine de l'image et notamment de rechercher des objets. Depuis quelques années, des approches dites spatiales essayent de fournir une description plus précise en considérant l'image comme composée d'un ensemble d'objets (de régions). La représentation de l'image est alors portée par l'ensemble des descriptions des éléments la composant mais aussi par les relations existantes entre eux. La conceptualisation de l'image a ainsi gagné un niveau. Nous détaillons dans un premier temps l'approche globale puis, l'approche spatiale.

2. Approche globale

Cette approche considère l'image dans son ensemble et la caractérise en utilisant des statistiques calculées sur l'image entière. Les différentes approches globales portent généralement sur trois critères : la couleur, la texture et la forme. Pour chacun d'eux, un grand nombre de descripteurs ont été développés. Ils peuvent ensuite être combinés pour obtenir un descripteur complet et robuste. Cet ensemble de statistiques est généralement appelé signature de l'image. Ensuite dans le cadre d'une recherche par l'exemple, la démarche est généralement la suivante : (voir figure 1)

- Phase préliminaire : calculer les descripteurs de chaque image de la base de données ;

- Phase en ligne : calculer les descripteurs de l'image requête ;
- Phase de recherche : rechercher les images proches de l'image requête dans l'espace du (des) descripteur(s) utilisé(s).

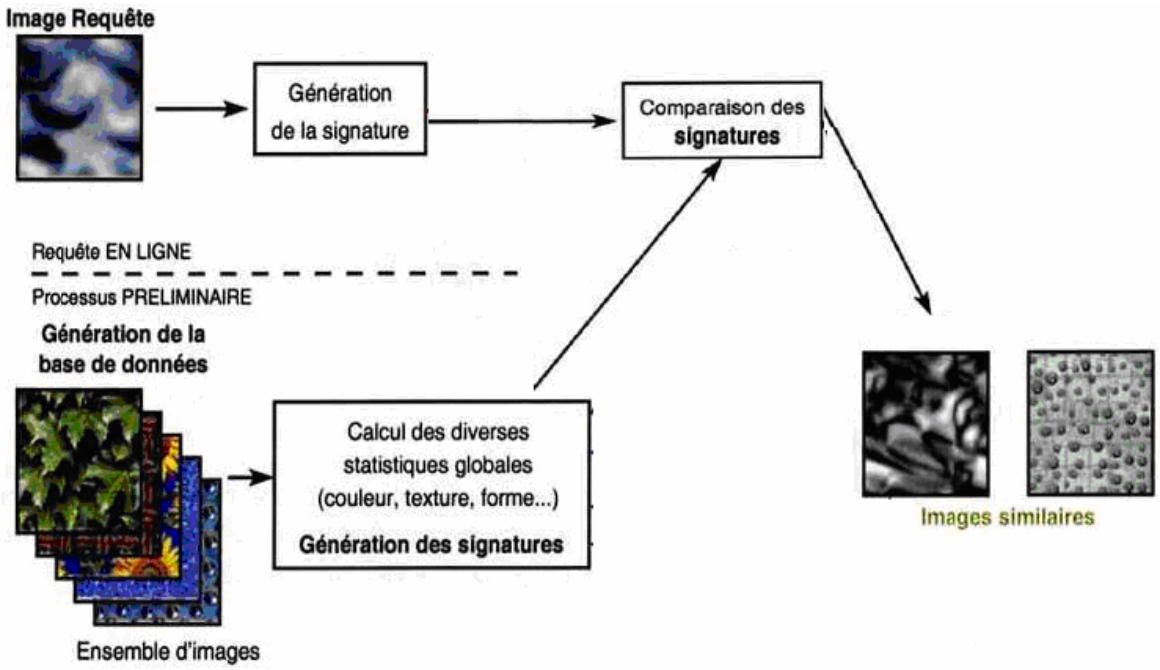


Figure 1. *Schéma d'indexation classique*

Dans ce qui suit on va détailler les trois critères de statistiques déjà cités.

2.1. Approche par analyse de l'histogramme de couleur

La couleur est sûrement le critère le plus important psycho visuellement parlant dans la vision d'une image. Les histogrammes de couleur (voir figure 2) sont largement répandus dans le domaine de l'indexation des images par le contenu. L'histogramme h d'une image (ou plus généralement, d'une région quelconque correspondant à un objet de forme arbitraire) est défini par les fréquences relatives d'apparition des couleurs. Les couleurs de l'image sont tout d'abord quantifiées en un nombre N de couleurs prototypes, notées $\{c_1, c_2,\ldots, c_N\}$, suivant les spécifications du descripteur de quantification. On a alors l'équation 1 :

$$\forall\, i \in \{1,2,\ldots, N\}, \qquad h(i) = \frac{\textit{Nombre de pixels ci}}{\textit{Nombre total de pixels}} \qquad (1)$$

Il est noté ici qu'avec cette définition l'histogramme obtenu est normalisé puisque $\sum h(i)=1$. Cependant l'inconvénient de l'histogramme de couleur est qu'il ne tient pas compte de la distribution spatiale de la couleur dans l'image

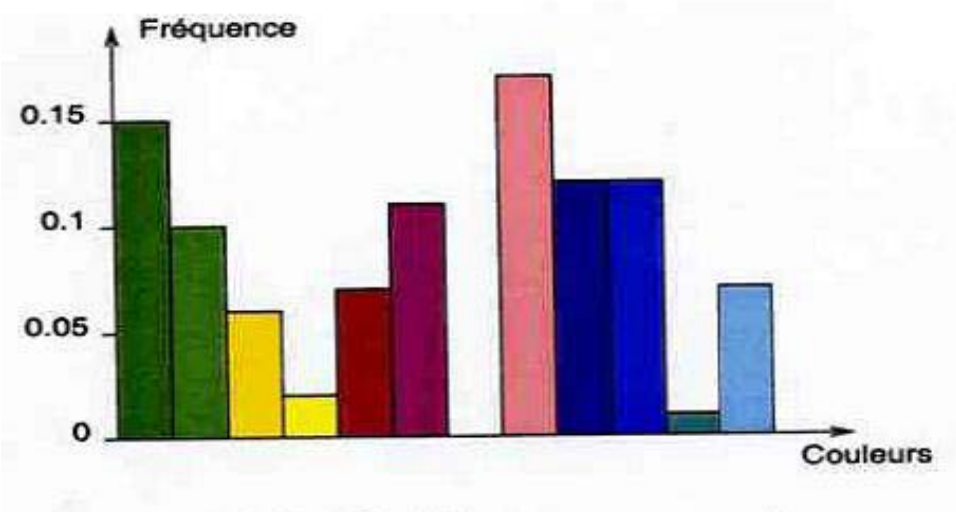


Figure 2. *Un histogramme couleur*

2.2. Approche par analyse de la texture

Les définitions académiques de la texture se restreignent souvent à des cas particuliers. L'une se réfère au tissage, une autre à la géologie. Cependant on retrouve toujours un critère important qui est la notion d'arrangement spatial. Dans le langage courant, le terme de texture est employé pour caractériser la surface d'un objet. On lui associe souvent un adjectif tel que texture granuleuse de la peau d'orange, texture lisse de la cerise. Pour reconnaître la texture d'un objet, l'homme dispose de deux sens : la vue et le toucher, mais ceux-ci n'apportent pas toujours la même information. Prenons l'exemple d'un mur granuleux que l'on photographie. Si on incruste cette photographie à son emplacement dans le mur réel, les informations issues de la vue restent identiques, celles issues du toucher deviennent différentes. Nous nous intéressons bien sûr ici uniquement à la partie visuelle. Une image est une représentation planaire d'une scène située en général dans un espace bidimensionnel. Elle est créée afin de proposer une entité observable par l'œil humain. Pour cela, l'information élémentaire qu'elle contient peut être transcrite en niveaux de gris, les caractéristiques de base de la texture les plus utilisées ont été définies par Amadasum, al et Rolland. Il s'agit du contraste, de la complexité, de la grossièreté, de la forme, de la direction et de la force.

La grossièreté : une texture grossière possède des primitives larges : il existe alors peu de variations entre l'intensité d'un pixel et celle des pixels voisins.

Le contraste : une texture possède un contraste élevé si les différences d'intensité entre primitives sont importantes.

La complexité : une texture complexe possède plusieurs types de primitives. Dans ce cas le contenu d'informations présentes dans la texture est important.

La force : plus la force est élevée et plus les primitives sont facilement définissables et visibles.

Plusieurs approches ont été élaborées pour détecter la répétition de motifs régulier ou irrégulier dans la totalité ou sur une région de l'image. Parmi ses approches nous trouvons celle basée sur les matrices de concurrence de niveaux de gris, d'autres se basant sur les décompositions à base d'ondelettes, d'autres se basant sur le filtre de Gabor.

2.3. Approche par analyse de la forme

La troisième classe de caractéristiques qui apparaît fréquemment dans les systèmes de recherche d'images est basée sur la forme.

Pour utiliser la forme comme une propriété d'image, il est essentiel de segmenter l'image pour détecter les objets ou les contours d'objets.

La forme peut être définie comme la description de l'objet sans tenir compte de son orientation, de sa position et de sa taille.

Ainsi, les descripteurs des formes sont invariants à la translation, à la rotation et à l'échelle.

On peut diviser les techniques de caractérisation des formes en deux approches.

- Les approches basées sur les régions. Par exemple les moments géométriques.
- Les approches basées sur les contours. Par exemple le descripteur de fourrier.

On va détailler ces deux approches.

2.3.1. Descripteur de forme basée sur les contours.

Durant plusieurs années, les méthodes se basant sur le contour ont été très utilisées vu que ces dernières captent le contour de l'image qui correspond à une caractéristique importante pour la perception humaine d'objet. De plus, ils ont une complexité moyenne et un temps de calcul relativement faible. Pour ces raisons, cette première méthode a donné de bons résultats pour quelques types d'applications telles que la reconnaissance de caractères, le codage d'objets, etc.

Le contour dans une image représente une forte variation de la valeur des pixels, la méthode à utiliser doit différencier entre les variations causées par le bruit et la texture et par

la variation des intensités des pixels. Nous trouvons dans la littérature trois approches pour la détection de contour d'une image :

- Méthodes différentielles (gradient, Laplacien).
- Méthodes par "Template" (Roberts, Prewit, Sobel, Kirsch).
- Méthodes par optimisation basées sur : modélisation de contour, bruits, mesure de qualité de détection… (Marr, Canny).

Les deux descripteurs de contour les plus utilisés sont : le descripteur de Fourrier et le descripteur CSS ("Curavature Scale Space") adoptés par le standard MPEG7 ("Moving Picture Experts Group 7").

En général, pour des images plus complexes, le contour ne suffit pas pour décrire le contenu de l'image. Pour y remédier, nous détaillerons la deuxième approche basée sur région. Cette deuxième technique prend en considération la répartition des pixels dans l'image et n'a pas besoin de connaître les points du contour. Cette approche fera l'objet du paragraphe suivant.

2.3.2. Descripteur de formes basé sur les régions.

Les descripteurs de formes basées sur les régions prennent en considération en plus de la frontière de la forme son intérieur. Les meilleurs descripteurs de cette catégorie sont celles qui sont basé sur les moments.

Hu [20], a fait le premier pas dans l'utilisation des moments invariants pour la reconnaissance des formes vers 1960 en proposant les six moments de Hu qui sont basés sur les moments géométriques. Par la suite beaucoup d'autres moments ont été proposés : ''Legender Moment'', ''Zernike Moment'', ''Pseudo-Zernike Moment'', ''Rotational Moment'' et ''Complex Moment''. Ces moments sont devenus de plus en plus populaires à cause de leur description compacte, leurs performances et la possibilité de choisir le niveau de détail à atteindre, ce qui explique leur présence dans diverses applications [8,23].

3. Approche spatiale

La problématique de l'indexation dite spatiale prend naissance vers la fin des années 1980 et se développe rapidement à la fin du siècle dernier.

L'idée pour cette approche est de considérer l'image comme un ensemble d'objets et non plus comme une entité unique. Une fois l'image segmentée en plusieurs régions, ces dernières peuvent être caractérisées de la même manière que les images dans la partie précédente.

Grace à cette approche des détails plus fins peuvent être retrouvés au sein des images car les statistiques sont localisées pour chaque région de l'image, ce qui induit une diminution du problème de masquage d'une donnée par une autre (la composition d'un objet ne sera plus noyée dans celle de l'image mais sera isolée). De plus, il est possible de rechercher des arrangements spécifiques d'objets en utilisant l'information liée à l'organisation des régions dans l'image.

Cette approche nous permet même d'effectuer des requêtes partielles. On peut alors chercher les images qui contiennent l'image requête par exemple.

En effet, dans le cadre des requêtes partielles, une fois les images segmentées et les régions décrites, le système de recherche est exactement le même que celui de l'approche classique, mais les éléments recherchés ne sont pas les images entières mais certaines régions particulières.

Ensuite dans le cadre d'une recherche par l'exemple, la démarche est généralement la suivante :

- Phase préliminaire : segmentation des images de la base de données et calcul des descripteurs pour chaque région pour obtenir les représentations des images.
- Phase en ligne : segmentation de l'image requête et calcul des descripteurs des régions pour obtenir la représentation de l'image. Les régions à utiliser pour la recherche peuvent aussi être sélectionnées.
- Phase de recherche : rechercher les images proches de l'image requête en comparant les descriptions.

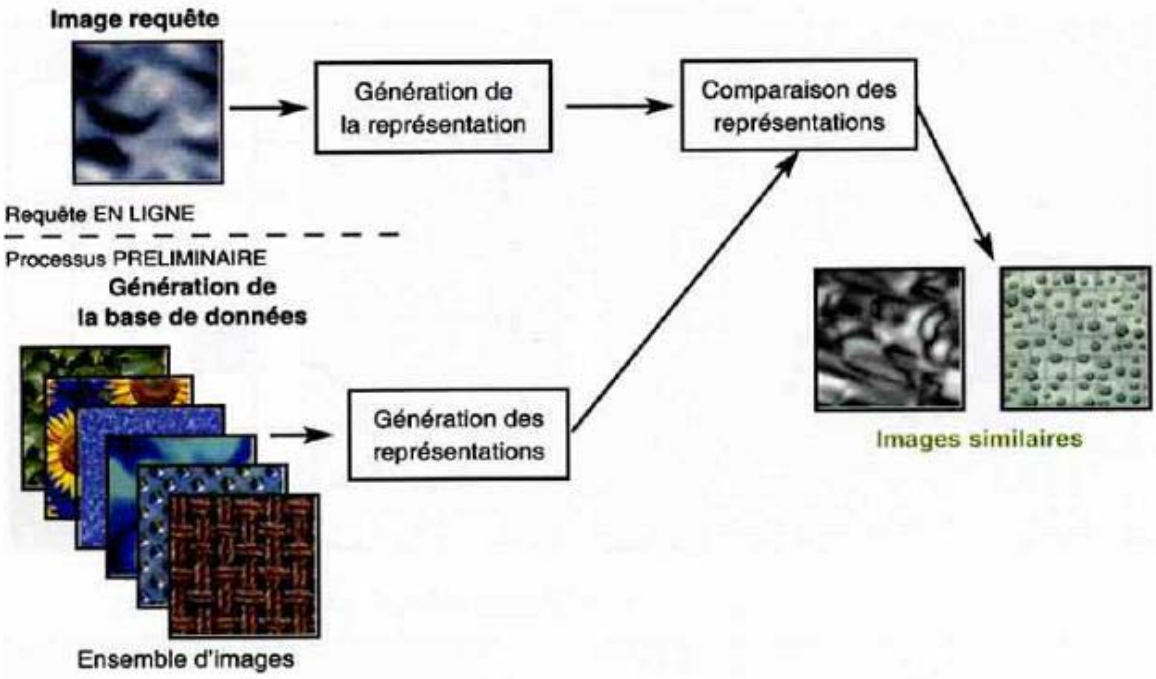


Figure 4. *Schéma d'indexation spatiale*

Les deux approches globales les plus puissantes pour la correspondance et la recherche par le contenu spatial des images sont les graphes relationnels attribués (Attribut Relation Graphe) et la projection symbolique (2D String).

C'est pourquoi nous allons détailler ces deux approches.

3.1. La projection symbolique (2D String).

La technique d'indexation symbolique d'images [11] est une nouvelle tendance dans le développement de la base de données d'images. Cette technique est basée sur une représentation efficace du contenu d'une image complexe par sa chaîne 2D. On affecte pour chaque objet un identifiant unique avant le stockage. Les positions relatives entre les objets sont alors représentées par deux chaînes de caractères mono- dimensionnelles. L'idée est de projeter la position des objets (i.e. leurs centres de gravité) sur les deux axes de l'image et de les prendre dans le même ordre que leurs projections. Le problème de recherche est alors transformé en une comparaison de chaînes des caractères.

La limite de 2 D String et de ses approches dérivées est de ne donner que des réponses de type binaire (oui / non) .Par conséquent il se peut qu'on ne trouve pas d'images similaires à notre image requête. En plus, cette approche devient inefficace dans le cas des régions chevauchantes (il n'ya pas une intersection entre les plus petits rectangles englobant les régions).

L'approche basée sur les graphes de relations attribués (ARG) représente l'image d'une façon plus détaillée que 2 D String. Ce qui lui permet de surmonter les problèmes de l'approche 2 D String.

Nous détaillons dans la section suivante l'approche ARG.

3.2. Les graphes relationnels attribués (ARG)

L'utilisation des graphes comme un outil mathématique pour résoudre des problèmes complexes comme celui de Königsberg remonte au 17ème siècle. Plus tard les graphes ont été utilisés par Kirchoff comme un outil de modélisation de circuit électrique (lois de Kirchoff: 1854). Récemment, la communauté de vision par ordinateur et de reconnaissance des formes a utilisé les graphes comme un outil aussi de modélisation et de recherche d'images (Barrow, Popplestone et Burstall [16] [17]. De nos jours, on assiste à un regain d'intérêt pour l'utilisation des graphes après avoir été abandonnés vu la complexité et la difficulté de trouver des algorithmes efficaces et rapides pour manipuler et utiliser les graphes. L'intérêt d'utiliser les graphes en analyse d'images se trouve certainement dans leur puissance à modéliser les dépendances entre objets présents sur une image et à tirer profit de la répartition spatiale de ces objets comme information potentielle pour décrire une image.

Un exemple de graphe construit à partir d'une image est représenté dans la figure 5

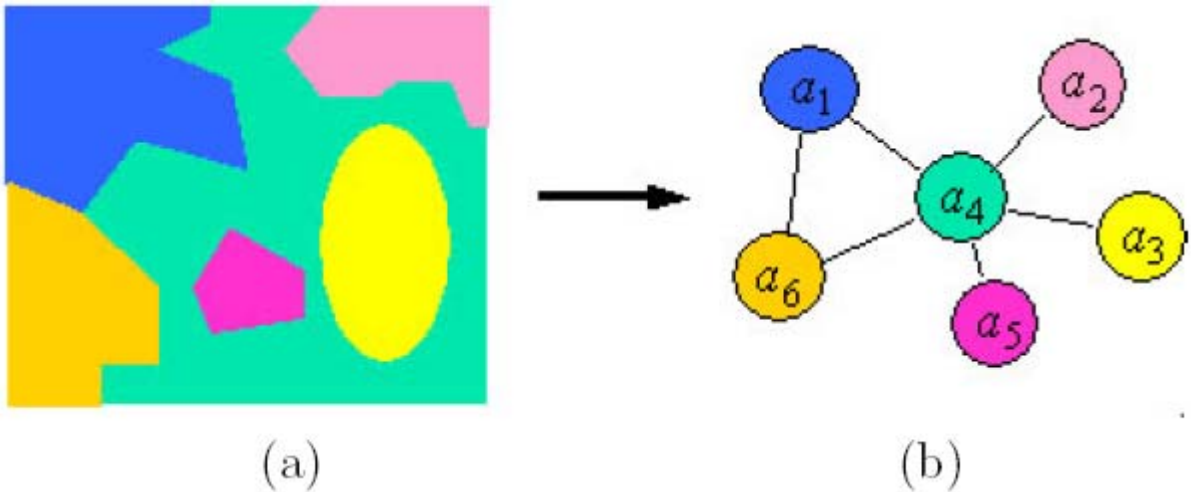


Figure 5. *Un graphe attribué qui représente les relations entre les régions de l'image.*
(a)Une image qui contient un groupe de régions.
(b)Le graphe qui représente (a).
La couleur d'un nœud est celle de la région qui lui correspond.
Les arcs représentent les relations d'adjacence entre les régions.

Le graphe modèle est souvent construit en utilisant un nœud pour chacune des régions de l'objet à reconnaitre (Ex: dans le cas d'images d'un cerveau humain, il y aura un nœud pour représenter le cervelet, un autre pour chaque ventricule...) et les arêtes pour représenter les relations entre ces régions. On utilise des attributs pour exprimer les propriétés de chaque nœud ou arête, et pour pouvoir les identifier ainsi que pour les distinguer les uns des autres. Ce graphe est souvent construit de manière supervisée.

Après la construction d'un graphe modèle, et pour pouvoir réaliser la reconnaissance d'images à travers lui, il est nécessaire de construire un graphe à partir de chacune des images à reconnaitre. Ces graphes, qui sont dénommés dans la littérature graphes de données ou graphes d'entrées, sont automatiquement construits par l'ordinateur sans l'assistance de l'utilisateur. Dans ce processus de construction du graphe de données correspondant à une image, une étape de segmentation de l'image précède la construction du graphe est fondamentale : le graphe de données est formé en utilisant un nœud pour représenter chacun des objets obtenus. Les mêmes attributs que ceux du graphe modèle sont calculés pour les nœuds et les arêtes du graphe de données. Ainsi la recherche d'une image représentée par un graphe est interprétée comme une recherche de similarité entre les graphes ou sous graphes qui représentent chaque image. Le calcul de similarité entre deux graphes d'un point de vue mathématique est réalisé par une recherche de morphisme de graphe. Ce type d'opération entre les graphes s'appelle appariement de graphes qui est un problème NP-complet.

4. Conclusion

Dans ce chapitre nous avons détaillé les deux approches d'indexations qui existent dans la littérature.

Une étude comparative [12] a été faite sur les deux approches précédemment énoncées. Nous avons relevé que :

- L'approche 2-D String est nettement plus rapide que l'approche ARG.
- L'approche ARG est beaucoup plus performante que l'approche 2-D String aux niveaux de rappel et de précision.
- L'approche ARG est extensible, elle permet de représenter n'importe quel type et nombre de propriétés des régions de l'image et des relations entre elles.

Nous avons adopté dans ce projet de fin d'études l'approche ARG vu la finesse et la maniabilité qu'elle apporte aux niveaux de représentation de l'image et son taux de performance élevé. La phase la plus difficile dans l'approche ARG est l'appariement de graphes qui consiste à calculer la similarité entre deux graphes.

Nous détaillons dans le chapitre suivant les méthodes proposées pour résoudre ce problème. Ensuite nous proposons un algorithme d'appariement de graphes.

Chapitre II

Algorithme proposé pour l'appariement de graphes

1. Introduction

Plusieurs algorithmes d'appariement de graphes ont vu le jour, particulièrement dans les trente dernières années [1, 9, 10, 22]. Ces algorithmes ont été conçus selon différentes philosophies, avec des outils différents et pour des buts différents.
Techniquement parlant, on peut diviser les algorithmes d'appariement de graphes en deux approches.

La première approche consiste en une exploration déterministe de l'espace d'exploration. Parmi ces techniques on peut citer : arbre de recherche, clic maximal, A*.

La deuxième approche consiste en une exploration non déterministe de l'espace d'exploration. On peut citer parmi les techniques utilisées dans cette approche **:** algorithme d'optimisation par colonie de fourmis, algorithme de recherche locale taboue réactive, les réseaux de neurones, la relaxation probabiliste.

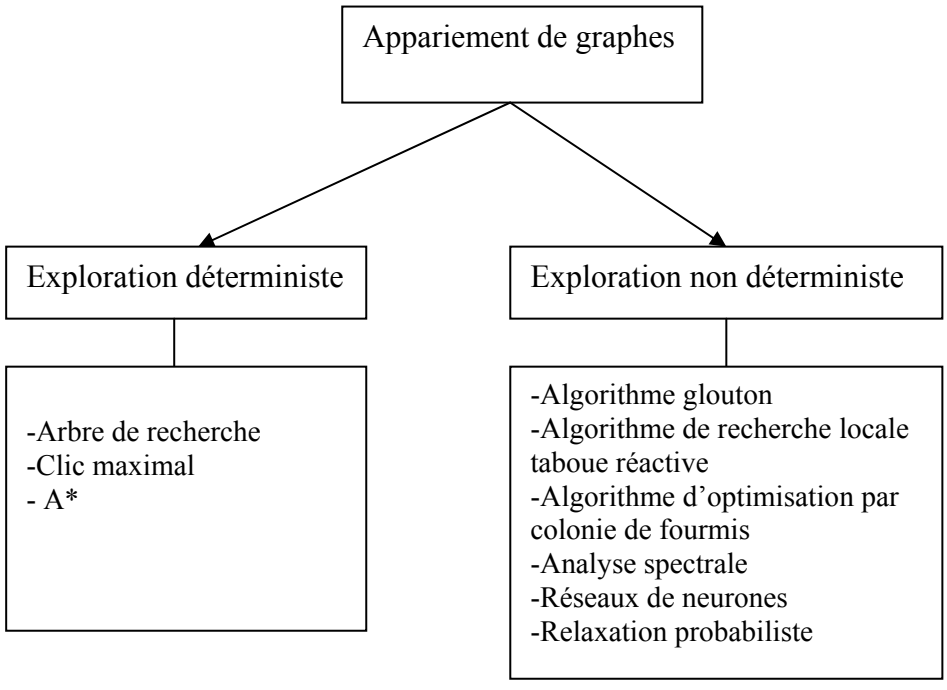


Figure 6. *Classification des algorithmes d'appariement de graphes*

Le but ultime de ces méthodes est de réduire la complexité de l'appariement de graphes en intégrant dans le processus uniquement deux graphes. L'appariement de graphes est réalisé par une recherche de l'isomorphisme de graphes, de sous-graphe ou du plus grand

commun sous graphe. Ces méthodes ont principalement utilisé deux approches : premièrement, une recherche arborescente déterministe utilisant des techniques heuristiques pour diriger le processus de recherche vers la solution du problème, deuxièmement, l'optimisation stochastique souvent employée dans la reconnaissance de formes statistiques.

Avant de détailler quelques méthodes d'appariement de graphes, nous présentons les principaux concepts et notions de base en appariement de graphes et en mesures de similarité entre les graphes

2. Notions de base

2.1. Notions de base : rappel et définitions

Dans cette section, nous présentons les principaux concepts et notions de base en appariement de graphes et de mesures de similarité entre les graphes.

2.1.1. Définition 1

On appelle graphe étiqueté, la structure $G = (S, A, \alpha, \beta)$, où

- S est un ensemble fini de sommets,
- A est un ensemble fini d'arêtes,
- $\alpha : S \to L_S$ est une fonction d'assignation d'étiquettes pour les sommets,
- $\beta : A \to L_A$ est une fonction d'assignation d'étiquettes pour les arêtes.

Où L_S et L_A sont deux ensembles d'étiquettes respectivement pour les sommets et les arêtes.

Exemple (Figure 7)

$G = (S, A, \alpha, \beta)$
$L_S = \{0.4, 0.3, 0.5\}$ $L_A = \{a, b, c\}$
$S = \{s_1, s_2, s_3\}$ $A = \{(s_1,s_2),(s_1,s_3),(s_2,s_3)\}$
$\alpha(s_1) = 0.5$ $\beta(s_1, s_2) = b$
$\alpha(s_2) = 0.4$ $\beta(s_1, s_3) = a$
$\alpha(s_3) = 0.3$ $\beta(s_2, s_3) = c$

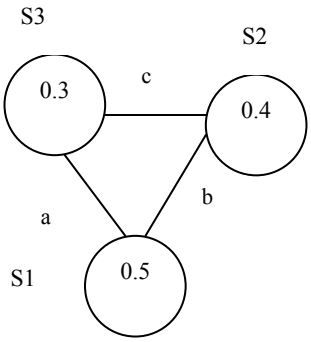


Figure 7. *Représentation graphique du graphe de l'exemple ci dessus*

2.1.2. Définition 2

Le graphe $G'=(S',A',\alpha',\beta')$ est dit un sous-graphe de $G=(S,A,\alpha,\beta)$

On note : $G'\subseteq G$ si et seulement si on a :

- $S'\subseteq S$

- $A'=A\cap(S'\times S')$

- $\alpha(s)=\alpha'(s)$ avec $s\in S'$

- $\beta(a)=\beta'(a)$ avec $a\in A'$

2.1.3. Définition 3

Étant donné deux graphes $G=(S,A,\alpha,\beta)$ et $G'=(S',A',\alpha',\beta')$.

Une application f:S $\rightarrow$ S' réalise un isomorphisme de graphes entre G et G' si et seulement si

- $|S|=|S'|$ et $|A|=|A'|$

- $\forall s'\in S',\exists! s\in S$, tel que $f(s)=s'$

- $\forall(s'_i,s'_j)\in A',\exists!(s_i,s_j)\in A$, tel que $f(s_i)=s'_i, f(s_j)=s'_j$

-L'isomorphisme de graphes peut être aussi réalisé en intégrant les deux fonctions d'assignement $\alpha(\alpha')et\beta(\beta')$

- $|S|=|S'|$ et $|A|=|A'|$

- $\forall s'\in S',\exists! s\in S$, tel que $f(\alpha(s))=\alpha'(s')$

- $\forall(s'_i,s'_j)\in A',\exists!(s_i,s_j)\in A$, tel que $\beta(f(s_i),f(s_j))=\beta'(s'_i,s'_j)$

La notion d'isomorphisme de sous-graphes est très proche de celle d'isomorphisme de

graphes. C'est une fonction injective de S dans S'.

Une application $f:S \to S'$ est un isomorphisme de sous-graphes entre G et G' si et seulement si

- $\exists S'' \subseteq S', \exists A'' \subseteq A'$, tel que la restriction de f aux graphes $G = (S, A, \alpha, \beta)$

et $G' = (S', A', \alpha', \beta')$ est un isomorphisme de graphes.

On distingue deux types d'isomorphisme : isomorphisme de graphes lorsque les deux graphes partagent le même nombre de sommets et d'arêtes et l'isomorphisme de sous-graphes dans le cas contraire. Un troisième type d'isomorphisme peut être distingué qu'on appelle le plus grand commun sous-graphe entre deux graphes. Ce dernier permet de trouver un isomorphisme de graphes entre deux sous-graphes (donc de même taille) appartenant à deux graphes donnés. Il s'agit du sous-graphe en commun possédant le plus grand nombre de sommets parmi tous les sous-graphes communs aux deux graphes.

Exemple

$G = (S, A, \alpha, \beta)$ $G' = (S', A', \alpha', \beta')$

$S = \{s_1, s_2, s_3\}$ $S' = \{s_1', s'_2, s'_3\}$

$A = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}$ $A' = \{(s'_1, s'_2), (s'_1, s'_3), (s'_2, s'_3)\}$

$LS = LS' = \{0.3, 0.4, 0.5\}$

$LA = LA' = \{a, b, c\}$

$\alpha(s_1) = 0.5$ $\beta(s_1, s_2) = b$ $\alpha'(s_1) = 0.3$ $\beta'(s'_1, s'_2) = c$

$\alpha(s_2) = 0.4$ $\beta(s_1, s_3) = a$ $\alpha'(s_2) = 0.4$ $\beta'(s'_1, s'_3) = b$

$\alpha(s_3) = 0.3$ $\beta(s_2, s_3) = c$ $\alpha'(s_3) = 0.5$ $\beta'(s'_2, s'_3) = a$

$f(s_1) = s'_3$ $f(s_2) = s_1'$ $f(s_3) = s_2'$

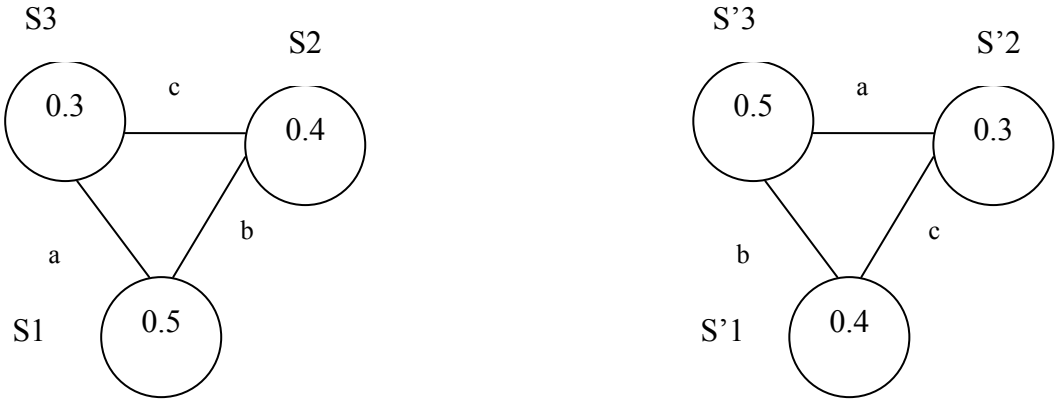


Figure 8. *Représentation graphique des deux graphes de l'exemple 2.1.3.1*

2.1.4. Définition 4

Soit G et G', deux graphes étiquetés. Le plus grand commun sous-graphe g entre G et G' est un sous-graphe à la fois de G et G' possédant le plus grand nombre de sommets parmi tous les sous-graphes isomorphes entre G et G'.

Exemple

$G = (S, A, \alpha, \beta)$ $G' = (S', A', \alpha', \beta')$

$S = \{s_1, s_2, s_3, s_4\}$ $S' = \{s'_1, s'_2, s'_3, s'_4\}$

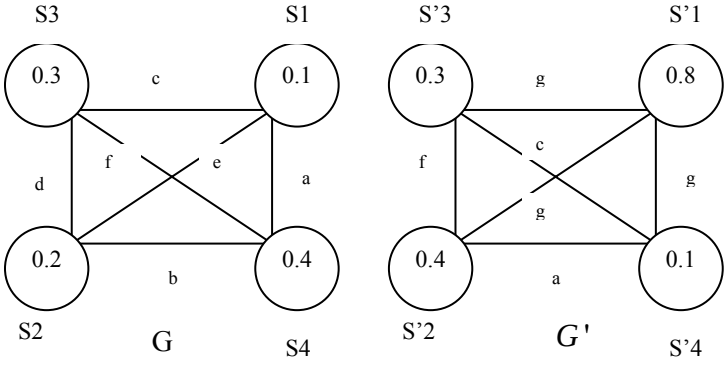
$A = \{(s_1, s_2), (s_1, s_3), (s_1, s_4), (s_2, s_3), (s_2, s_4), (s_3, s_4)\}$

$A' = \{(s'_1, s'_2), (s'_1, s'_3), (s'_1, s'_4), (s'_2, s'_3), (s'_2, s'_4), (s'_3, s'_4)\}$

$LS = LS' = \{0.1, 0.2, 0.3, 0.4, 0.8\}$

$LA = LA' = \{a, b, c, d, e, f, g\}$

	$\beta(s_1, s_2) = e$		$\beta'(s'_1, s'_2) = g$
$\alpha(s_1) = 0.1$	$\beta(s_1, s_3) = c$	$\alpha'(s_1) = 0.8$	$\beta'(s'_1, s'_3) = g$
$\alpha(s_2) = 0.2$	$\beta(s_1, s_4) = a$	$\alpha'(s_2) = 0.4$	$\beta'(s'_1, s'_4) = g$
$\alpha(s_3) = 0.3$	$\beta(s_2, s_3) = d$	$\alpha'(s_3) = 0.3$	$\beta'(s'_2, s'_3) = f$
$\alpha(s_4) = 0.4$	$\beta(s_2, s_4) = b$	$\alpha'(s_4) = 0.1$	$\beta'(s'_2, s'_4) = a$
	$\beta(s_3, s_4) = f$		$\beta'(s'_3, s'_4) = c$



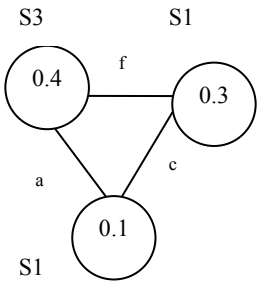


Figure 9. *Représentation graphique des trois graphes G et G' et leur plus grand commun sous-graphe g*

2.1.5. Définition 5

On appelle une opération d'édition une transformation d'un sommet ou d'une arête d'un graphe $G = (S, A, \alpha, \beta)$.

On définit six types d'opérations d'édition

1. substitution de sommet $\delta_{ss}(s_i)$
2. substitution d'arête $\delta_{sa}(s_i, s_j)$
3. suppression de sommet $\delta_{ds}(s_i)$
4. suppression d'arête $\delta_{ds}(s_i, s_j)$
5. insertion de sommet $\delta_{is}(s_i)$
6. insertion d'arête $\delta_{is}(s_i, s_j)$

Exemple

$G = (S, A, \alpha, \beta)$ $G' = (S', A', \alpha', \beta')$

$LS = \{0.4, 0.3, 0.5\}$ $LS' = \{0.4, 0.3, 0.5\}$

$LA' = \{a, b, c\}$

$S = \{s_1, s_2, s_3\}$ $S' = \{s_1', s'_2, s'_3\}$

$A = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}$ $A' = \{(s'_1, s'_2), (s'_1, s'_3), (s'_2, s'_3)\}$

$\alpha(s_1) = 0.5$ $\beta(s_1, s_2) = b$ $\alpha'(s'_1) = 0.4$ $\beta'(s'_1, s'_2) = c$

$\alpha(s_2) = 0.4$ $\beta(s_1, s_3) = a$ $\alpha'(s'_2) = 0.3$ $\beta'(s'_1, s'_3) = b$

$\alpha(s_3) = 0.3$ $\beta(s_2, s_3) = c$ $\alpha'(s'_3) = 0.5$ $\beta'(s'_2, s'_3) = a$

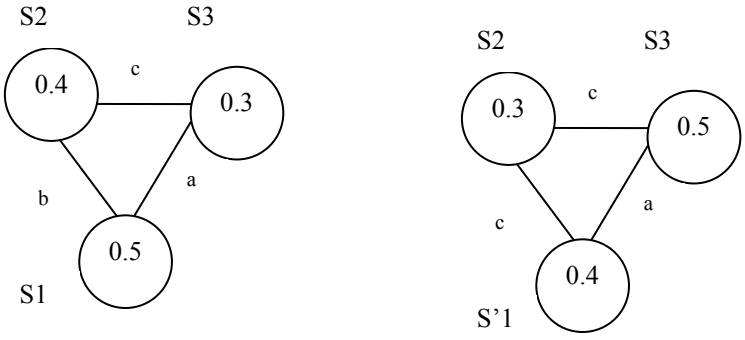


Figure 10. Représentation graphique des deux graphes de l'exemple ci dessus
Substitution du sommet S_1 : $\delta_{ss}(S_1) = S'_1$; Substitution de l'arête (S_1, S_2) :

$$\delta_{sa}(S_1, S_2) = (S'_1, S'_2)$$

2.1.6. Définition 6

On appelle une séquence d'édition un ensemble de transformations ordonnées $\delta_t^1,...\delta_t^{|\Delta|}$ transformant un graphe G en un graphe G' avec $t \in \{ss, sa, is, ia, ds, da\}$

Exemple

$\Delta = \{\delta_{ss}^1, \delta_{sa}^2\}$ avec $\Delta(G) = G''$

G et G'' les deux graphes représentés dans l'exemple 2.5

2.1.7. Définition 7

On appelle f_c une fonction coût permettant d'associer pour chaque opération d'édition appartenant à Δ un nombre réel positif.

$f_c : \Delta \to R$

Exemple

$f_c(\delta_{ss}^1) = d(0.5, 0.4) = 0.1$

$f_c(\delta_{sa}^2) = D(b, c) = 1$

d est la distance Euclidienne tandis que D est la différence entre l'ordre de a et de b dans l'alphabet.

2.1.8. Définition 8

Étant donné deux graphes $G_1 = (S_1, A_1, \alpha_1, \beta_1)$ et $G_2 = (S_2, A_2, \alpha_2, \beta_2)$, on appelle une erreur de correction d'appariement de graphes (ecag) de G_1 à G_2 , la fonction bijective $f : \overline{S_1} \to \overline{S_2}$ où $\overline{S_1} \subseteq S_1$ et $\overline{S_2} \subseteq S_2$.

La fonction f peut être interprétée comme une transformation du graphe G_1 en G_2 , en appliquant une séquence d'édition Δ.

Le coût d'une séquence d'édition Δ relatif à une ecag est défini par

$$C(\Delta)_{ecag} = \sum_{i=1}^{|\Delta|} f_c(\delta_t^i) \text{ avec } t \in \{ss, sa; is, ia, ds, da\}$$

La fonction f est une ecag optimale (ecago) si et seulement si, il n'existe aucune autre ecag f' transformant G_1 en G_2 dont le coût est inférieur à celui de f .

Exemple

Soit deux graphes $G=(S,A,\alpha,\beta)$ et $G'=(S',A',\alpha',\beta')$

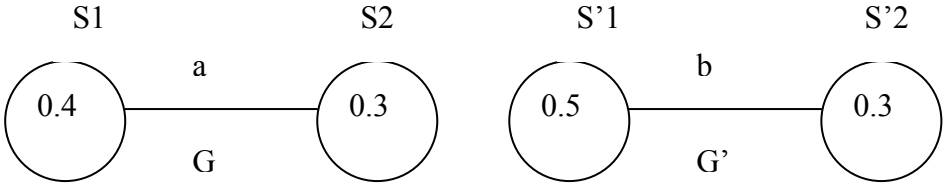


Figure 11. *Représentation graphique des deux graphes*

Avec $S=\{s_1,s_2\}$ et $S'=\{s'_1,s'_2\}$

f_1 est un ecag entre G et G' tel que

$f_1 : s_1 \to s'_1$ et $f_1 : s_2 \to s'_2$.

La séquence d'opération d'édition relative à f_1 est $\Delta=\{\delta^1_{ss},\delta^2_{ss},\delta^3_{sa}\}$

Avec $\delta^1_{ss}(s_1)=s'_1 \quad \delta^2_{ss}(s_2)=s'_2 \quad \delta^3_{sa}(s_1,s_2)=(s'_1,s'_2)$

Le coût associé à f_1 est $C(\Delta)_{f_1}=f_c(\delta^1_{ss})+f_c(\delta^2_{ss})+f_c(\delta^3_{sa})$

$f_c(\delta^1_{ss})=\mathrm{d}(0.4,0.5)=0.1$
$f_c(\delta^2_{ss})=\mathrm{d}(0.3,0.3)=0$
$f_c(\delta^3_{sa})=\mathrm{D}(\mathrm{a},\mathrm{b})=1$
$C(\Delta)_{f_1}=1.1$

3. Quelques méthodes d'appariement de graphes

3.1. Recherche arborescente par rétropropagation

Cette méthode consiste en une recherche arborescente où chaque sommet de l'arbre représente un isomorphisme de sous-graphes partiel et une arête et son extension par une nouvelle mise en correspondance. La recherche de l'isomorphisme est conduite par l'application d'une séquence d'opérations d'édition notée Δ enchaînée sur un graphe G dont le coût global noté $C(\Delta)$ est minimal. Le but principal est de trouver une séquence d'opérations d'édition notée Δ transformant un graphe G en un graphe G' dont le coût $C(\Delta)$ est minimal.

Le processus de recherche décrit ci-dessus permet de trouver l'erreur de correction d'appariement de graphes optimale. Cependant, afin de trouver la solution optimale, le processus doit effectuer $O(|S'|^{|S|})$ mises en correspondance entre les sommets. Pour chaque mise en correspondance, le processus doit effectuer $O(|S|^2)$ mises en correspondance entre les arêtes. Ainsi $O(|S|^2|S'|^{|S|})$ opérations sont nécessaires pour déterminer la solution optimale. La progression du processus de recherche dont nous avons rendu compte dans la description de l'algorithme est basée sur le coût minimal de l'appariement. La performance de l'algorithme peut être améliorée si d'autres indicateurs sont utilisés pour mieux renseigner l'algorithme et progresser dans la bonne voie, ce qui réduit considérablement le temps nécessaire pour rechercher la solution optimale. L'algorithme de recherche *A**, par le recours à la notion du coût futur permet d'informer le processus de recherche sur l'état qui doit être développé en premier ou le chemin à emprunter. Une méthode d'estimation du coût futur efficace a été proposée par Wong [13]. L'idée de base est d'estimer pour chaque appariement p un coût seuil. Chaque sommet n'appartenant pas à p est donc apparié à un sommet du deuxième graphe qui lui non plus n'appartient pas à p. Le coût de chaque opération de substitution de sommet ainsi que le coût de chaque opération de substitution d'arête relative à la mise en correspondance de sommet sont calculés pour former un coût minimum représentant un coût seuil (il doit être ajouté au coût réel de l'appariement p. Il est clair que la complexité du problème n'est pas réduite par l'estimation du coût futur. Au contraire il faut $O(|S|^2|S'|^{|S|+1})$ opérations pour trouver la solution optimale. Cependant, le processus de recherche sera bien informé et se dirigera vers la solution optimale plus rapidement qu'une recherche arborescente sans estimation du coût futur.

3.2. Algorithme d'optimisation par colonie de fourmis.

Le principe des algorithmes d'optimisation à base de colonie de fourmis (ACO) [9,10] consiste à reformuler le problème à résoudre en un problème de recherche d'un chemin optimal dans un graphe (appelé graphe de construction) et à utiliser des fourmis artificielles pour trouver les bons chemins de ce graphe. A chaque itération de l'algorithme (appelée cycle), chaque fourmi de la colonie construit aléatoirement un chemin du graphe (*i.e.*, une solution du problème) et de la phéromone est déposée sur les meilleurs chemins découverts lors de ce cycle.

Lors des cycles suivants, les fourmis construisent de nouveaux chemins avec une probabilité dépendant de la phéromone déposée lors des cycles précédents et d'une heuristique propre au problème considéré. La colonie de fourmis converge alors peu à peu vers les meilleures solutions. Des mécanismes de contrôle de la quantité de phéromone déposée sur le graphe de construction déterminent l'efficacité d'un algorithme ACO : la quantité de phéromone déposée sur les composants des solutions peut être bornée afin de réguler l'effort de diversification de la recherche et une évaporation de la phéromone au cours du temps est mise en place afin "d'oublier" les mauvaises solutions construites.

3.3. Algorithme glouton.

Un algorithme glouton de recherche du meilleur appariement de deux graphes est proposé [5]. L'algorithme démarre d'un appariement vide (*i.e.*, aucun sommet n'est apparié). A chaque itération, un nouveau couple de sommets est ajouté à l'appariement courant. Le couple ajouté est choisi aléatoirement parmi ceux dont l'ajout maximise la similarité courante. Ce processus est itéré tant que l'ajout d'un couple à l'appariement améliore sa similarité. Cet algorithme a une complexité faiblement polynomiale et, comme il est non déterministe, il peut être exécuté plusieurs fois afin de garder le meilleur appariement trouvé.

3.4. Algorithme de recherche locale Taboue réactive.

L'algorithme glouton retourne un appariement M "localement optimal" : ajouter ou enlever un couple de sommets à M ne peut l'améliorer. Cependant, il est parfois possible d'améliorer M en lui ajoutant et/ou en lui supprimant plusieurs couples de sommets. Une recherche locale cherche à améliorer une solution courante en explorant son voisinage : les voisins d'un appariement M sont les appariements obtenus en ajoutant ou en supprimant un seul couple de sommets à M. En démarrant d'un appariement initial, une recherche locale explore l'espace de recherche en se déplaçant de voisin en voisin jusqu'à l'obtention de la solution optimale ou l'utilisation du nombre maximum d'itérations autorisé. A chaque itération, le prochain voisin à explorer est choisi selon une heuristique. La recherche Taboue [15,21] est une des meilleures heuristiques connues pour le choix du prochain voisin. Le voisin qui maximise la similarité est toujours sélectionné mais, afin de ne pas rester autour d'un maximum local en réalisant toujours les mêmes mouvements, une liste Taboue est utilisée. Cette liste mémorise les k derniers mouvements réalisés (*i.e.*, les k derniers couples de sommets ajoutés ou enlevés) afin d'interdire les mouvements inverses durant k itérations

(*i.e.*, ajouter un couple de sommets précédemment supprimé ou supprimer un couple de sommets précédemment ajouté). Ce mécanisme permet de s'échapper des maxima locaux.
La longueur k de la liste taboue est un paramètre critique : si la liste est trop longue, l'algorithme converge trop lentement, et si elle est trop courte, l'algorithme est incapable de quitter les maxima locaux et d'améliorer la solution.

3.5. Synthèse

Parmi les algorithmes d'appariement de graphes déjà cités dans la section précédente, l'algorithme de recherche arborescente par rétropropagation est le seul qui permet de retrouver avec certitude la solution optimale. Le problème de cet algorithme est sa complexité très élevée.
Dans le paragraphe suivant nous allons présenter un algorithme de complexité réduite qui permet néanmoins de fournir une solution optimale dans la majorité des cas.

4. Algorithme proposé

4.1. Introduction

Dans cette section, nous proposons un algorithme d'appariement de graphes. L'algorithme développé permet de trouver une solution approximative, souvent très proche de l'optimale. L'appariement trouvé représente un isomorphisme de graphes ou de sous-graphes entre deux graphes donnés, en fixant le nombre de précision K. Le paramètre *K* indique le nombre de sommets qui peut être utilisé pour trouver l'isomorphisme. La valeur de *K* peut varier entre 1 et le nombre de sommets dans le plus grand graphe. L'efficacité de l'algorithme résulte de la faible valeur de *K* utilisée dans l'algorithme, ce qui réduit d'une manière considérable l'espace de recherche. Dans le paragraphe suivant, nous allons exposer notre algorithme qui calcule de manière intelligente la distance entre le graphe en évitant de calculer les 'mauvais' isomorphismes.

4.2. Appariement de graphes basé sur la mise en correspondance des sommets

Ce type d'appariement de graphes consiste à chercher des mises en correspondance entre les sommets d'un premier graphe et les sommets d'un deuxième graphe, d'une part, et des mises en correspondance entre les arêtes des deux graphes, d'autre part [1]. Un appariement de graphes généré par application d'opérations d'édition est interprété comme une distance (ecag) entre les deux graphes mis en correspondance. Notre but est de rechercher l'appariement dont la distance est la plus faible. Une première solution consiste à minimiser cette distance en intégrant dans le processus de recherche à la fois les mises en correspondance des sommets et les mises en correspondance des arêtes. Malheureusement cette idée n'est pas en mesure de donner des résultats même pour des graphes de faible taille. Le processus de minimisation est un problème difficile. Le temps nécessaire pour trouver une solution croît exponentiellement en fonction de la taille des deux graphes. Une solution approximative s'impose pour éviter la complexité élevée du problème. L'idée de base de notre algorithme repose sur une séparation entre la mise en correspondance des sommets et la mise en correspondance des arêtes. Afin de réduire l'espace de recherche, la recherche de l'appariement est réalisée en minimisant tout d'abord le coût des mises en correspondance des sommets. L'algorithme procède par une exploration de l'espace de recherche en deux phases. L'hypothèse utilisée dans l'algorithme consiste à rechercher la mise en correspondance des sommets en première phase, suivie d'une mise en correspondance entre les arêtes des deux graphes en deuxième phase. Intuitivement, cette hypothèse est très raisonnable dans les applications pratiques, vu que la similarité entre deux objets en premier lieu est une similarité entre leur contenu suivi d'une similarité spatiale des objets.

Etant donné deux graphes $G = (S, A, \alpha, \beta)$ et $G' = (S', A', \alpha', \beta')$, l'algorithme proposé dans ce projet de fin d'études permet de trouver l'ecag en deux phases d'itération. Ici, on suppose que le graphe G est plus petit ou égal à G' en terme de nombre de sommets, c'est-à-dire $|S| \leq |S'|$. En utilisant une valeur très grande pour *K,* l'algorithme est en mesure de trouver *l'ecago, l'erreur de correction de l'appariement de graphes optimal.* Il faut signaler que la valeur de *K* ne peut jamais dépasser en aucun cas la valeur de $|S|$. Cependant, l'algorithme est capable de trouver de très bons appariements (souvent optimaux) en utilisant une valeur de *K* beaucoup moins grande que $|S|$ (ex. *K* = 3, *K* = 5, etc.). Ce qui est un avantage considérable, permettant

à la fois de ne pas explorer tout l'espace de recherche et d'introduire une flexibilité sur la qualité du résultat attendu. Le processus d'appariement est itératif et se déroule comme suit. Dans la première phase, l'algorithme examine les appariements qui comportent les meilleures mises en correspondance des sommets. Par "meilleur", on entend les mises en correspondance dont la valeur est la plus faible parmi l'ensemble des mises en correspondance. L'ensemble des mises en correspondance entre deux graphes comporte $|S| \times |S'|$ éléments. A la fin de cette phase tous les isomorphismes qui comportent probablement la solution optimale sont retenus. Dans la deuxième phase, l'algorithme calcul les ecag à partir des mises en correspondance déjà établis dans la phase 1. L'ecago est probablement le plus petit ecag calculé dans cette phase.

4.3. Description détaillée de l'algorithme

Formellement l'algorithme est décrit de la manière suivante : étant donné deux graphes $G = (S, A, \alpha, \beta)$ et $G' = (S', A', \alpha', \beta')$, une matrice (Pij) de dimensions $|S| \times |S'|$ est introduite. Chaque élément ou mise en correspondance Pij dans P représente la similarité entre le sommet i du premier graphe et le sommet j du deuxième graphe. Une deuxième matrice binaire $B = (bij)$ de dimension $|S| \times |S'|$ est introduite afin d'identifier le ou les mises en correspondance les plus prometteuses qui formeront le ou les différents appariements. L'algorithme commence par l'initialisation de la matrice P suivie de celle de B. Chaque élément Pij de la matrice P est initialisé à $Pij = d(\alpha(si), \alpha'(s'j))$. La matrice B est ensuite construite de façon à contenir pour chaque ligne i K éléments de valeur 1.Ces éléments correspondent aux K plus proches sommets de G' au sommet i de G. A partir de la matrice B on détermine tous les appariements valides. Un appariement est considéré valide si et seulement si il représente un isomorphisme de graphes ou de sous-graphes. La phase de détermination des appariements valides est délicate. Nous allons présenter par la suite un exemple pour comprendre le fonctionnement de notre algorithme.Par la suite pour chaque appariement valide on calcule l'erreur engendrée par les mises en correspondance des sommets. L'erreur trouvée est additionnée à l'erreur engendrée par les mises en correspondance des arêtes. Cette erreur calculée représente l'ecag. A la fin nous déduisons la distance entre les deux graphes G et G' qui représente l'erreur de correction de l'appariement du graphes optimal (ecago). L'ecago représente la plus petite ecag calculée. Nous présentons le pseudo algorithme de l'appariement de graphes :

Procédure : Appariement_Sommets_Base (K, G, G')

Entrées

1) Deux graphes $G = (S, A, \alpha, \beta)$ et $G' = (S', A', \alpha', \beta')$

2) La précision K

Sortie

1) Les appariements valides.

Début

1. Initialiser P

Pour chaque i, j avec $1 < i < |S|$ et $1 < j < |S'|$ faire $Pij = d(\alpha(si), \alpha'(s' j))$

2. Initialiser B

Pour chaque i, j avec $1 < i < |S|$ et $1 < j < |S'|$

Si Pij est parmi les K plus petits éléments de la ligne i de P

Alors $Bij = 1$

Sinon $Bij = 0$

3. Initialiser B'

Pour chaque i, j avec $1 < i < |S|$ et $1 < j < |K|$

B'ij=le numéro de la colonne des éléments de B de la ligne i qui ont 1 comme valeur.

4. Construire les appariements valides à partir de B'

Fin

Procédure : Evaluation_appariement (V, G, G')

Entrées

1) Vecteur V contenant les appariements valides

2) Deux graphes $G = (S, A, \alpha, \beta)$ et $G' = (S', A', \alpha', \beta')$

Sortie

1) Le meilleur appariement v.

Début

Pour chaque appariement v

Calculer ecags

Calculer ecaga

ecag= acags+ecaga

Mettre à jour le meilleur appariement.

Fin

4.4. Exemple

Dans cette section nous présentons un exemple pour détailler dans un premier temps la démarche adoptée pour la recherche de l'isomorphisme de sous graphes selon l'algorithme présenté. Et dans un deuxième temps la démarche pour le calcul de l'ecago.

La figure ci-dessous (Figure 12) présente deux graphes G et G'. Le premier et le deuxième graphe comportent respectivement trois et quatre sommets. Le nombre à l'intérieur du cercle représente l'étiquette du sommet tandis que celui à l'extérieur représente l'ordre du sommet. Le nombre à coté de chaque ligne reliant deux sommets représente l'étiquette de l'arête.

$G = (S, A, \alpha, \beta)$ $G' = (S', A', \alpha', \beta')$

$S = \{s_1, s_2, s_3\}$ $S' = \{s_1', s'_2, s'_3, s'_4\}$

$A = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}$ $A' = \{(s'_1, s'_2), (s'_1, s'_3), (s'_1, s'_4), (s'_2, s'_3), (s'_2, s'_4), (s'_3, s'_4)\}$

$LS = LS' = \{0.3, 0.5, 0.6, 0.8, 0.9\}$

$LA = LA' = \{0.1, 0.2, 0.4, 0.5, 0.6, 0.7, 0.8\}$

$\alpha(s_1) = 0.3$ $\beta(s_1, s_2) = 0.2$

$\alpha(s_2) = 0.6$ $\beta(s_1, s_3) = 0.7$

$\alpha(s_3) = 0.9$ $\beta(s_2, s_3) = 0.4$

$\alpha'(s'_1) = 0.9$ $\beta'(s'_1, s'_2) = 0.5$

$\alpha'(s'_2) = 0.5$ $\beta'(s'_1, s'_3) = 0.1$

$\alpha'(s'_3) = 0.2$ $\beta'(s'_1, s'_4) = 0.8$

$\alpha'(s'_4) = 0.3$ $\beta'(s'_2, s'_3) = 0.6$

$\beta'(s'_2, s'_4) = 0.7$

$\beta'(s'_3, s'_4) = 0.3$

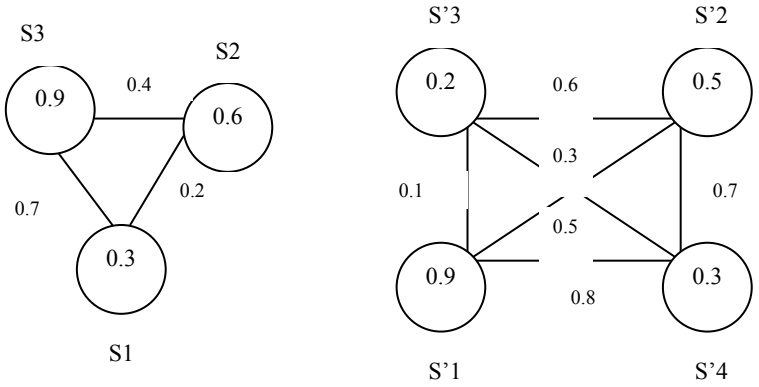


Figure 12. *Représentation graphique des deux graphes G et G'*

La première étape consiste à construire la matrice P.

	S'1	S'2	S'3	S'4
S1	0.6	0.2	0.1	0
S2	0.3	0.1	0.4	0.3
S3	0	0.4	0.7	0.6

Matrice P

En supposant que la précision est égale à 3.

La matrice B obtenue est la suivante :

	S'1	S'2	S'3	S'4
S1	0	1	1	1
S2	1	1	0	1
S3	1	1	0	1

Matrice B

La matrice B' déduite à partir de B obtenue est la suivante

2	3	4
1	2	4
1	2	4

Matrice B'

A partir de la matrice B' on obtient $3^3 = 27$ combinaisons possibles.

A partir de ces 27 combinaisons, on doit extraire seulement celles qui sont valides.

Il faut donc extraire seulement les combinaisons qui correspondent à un isomorphisme de graphes valide.

La combinaison 2 1 1 correspond à l'isomorphisme de sous-graphes $f:S \rightarrow S'$ entre G et G' tel que :

$f(s_1) = (s'_2)$

$f(s_2) = (s'_1)$

$f(s_3) = (s'_1)$

Pour que l'isomorphisme soit valide, chaque nœud de G doit lui correspondre un et un seul nœud de G' et l'inverse est vrai. L'application f doit être par conséquent injective.

Pour la combinaison 2 1 1, le nœud s'_1 lui correspond le nœud s_2 et s_3. Par conséquent cet isomorphisme n'est pas valide.

Ainsi, pour avoir un isomorphisme de sous-graphes valide entre G et G', la combinaison doit avoir des valeurs distinctes deux à deux.

La combinaison 2 1 4 correspond à l'isomorphisme de sous-graphes $f:S \rightarrow S'$ entre G et G' tel que :

$f(s_1) = (s'_2)$
$f(s_2) = (s'_1)$
$f(s_3) = (s'_4)$

Cette combinaison correspond à un isomorphisme de sous-graphes valide.

Dans notre exemple l'ensemble des combinaisons qui forme des isomorphismes valides sont :

2 1 4 2 4 1 3 1 2 3 1 4 3 2 1

3 2 4 3 4 1 3 4 2 4 1 2 4 2 1

Après avoir déterminé tous les isomorphismes de sous-graphes entre G et G' valides, on passe à l'étape de calcul des erreurs engendrées par chaque appariement de graphes.

Pour mieux comprendre le processus de calcul de l'ecag (erreur de correspondance de l'appariement de graphes), nous allons le calculer pour la combinaison 2 1 4.

Afin de faciliter la tâche, on va effectuer une transformation du graphe G' pour obtenir le graphe G'' à partir de cette combinaison. (Voir figure 13)

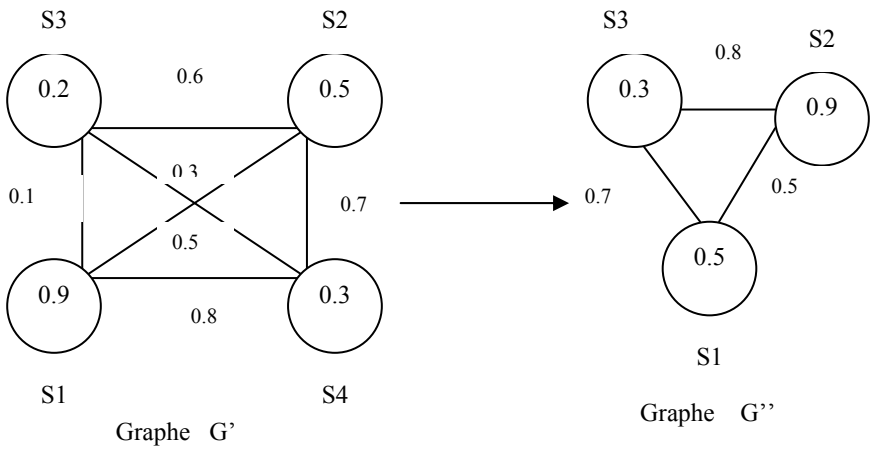


Figure 13. *Construction du graphe G'' à partir de la transformation de G' à partir de la combinaison 2 1 4*

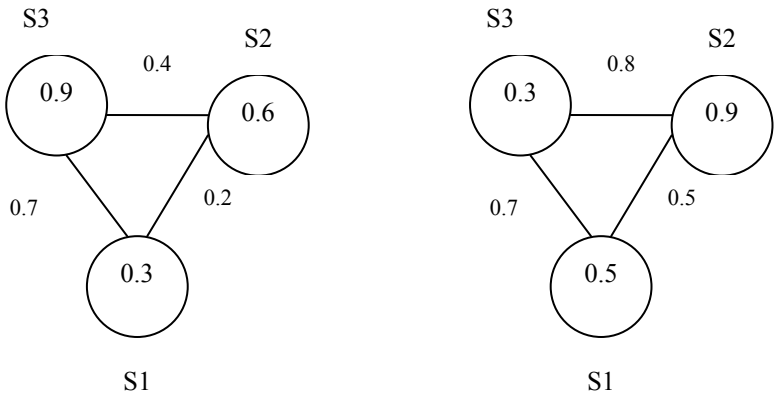


Figure 14. *Représentation graphique des deux graphes G et G''*

Après cette transformation l'ecag entre G et G' n'est que la différence terme à terme des sommets et des arcs du nouveau graphe obtenue G'' avec le graphe G.

ecag=ecags + ecaga

Avec ecags : erreur de mise en correspondance des sommets

ecaga : erreur de mise en correspondance des arêtes

Dans cet exemple :

ecags= 0.2+0.3+0.6=1.1

ecaga= 0.3+0+0.4=0.7

D'où pour cet appariement ecag=1.8

Après avoir calculé tous les ecag à partir des appariements valides, on peut déduire la distance entre les deux graphes qui représente le plus petit ecag calculé (ecago).

5. Conclusion

Nous avons proposé dans ce chapitre un algorithme d'appariement de graphes. L'algorithme permet d'extraire l'isomorphisme de graphes ou de sous-graphes. L'algorithme repose sur la notion d'opérateurs d'édition pour trouver l'isomorphisme. Trois types d'opérateurs peuvent être utilisés pour transformer un graphe en un autre. La substitution, la suppression et l'insertion. L'idée de base de notre algorithme repose sur une séparation entre la mise en correspondance des sommets et la mise en correspondance des arêtes. Le meilleur appariement est extrait. Ensuite, il est évalué et finalement comparé au meilleur de tout le processus. Nous avons présenté une version traitant l'opération de substitution et de suppression seulement. Une extension généralisant l'algorithme sur toutes les autres

opérations d'édition peut être proposée au futur. L'algorithme développé offre de nouvelles techniques pour rediriger rapidement le processus de recherche vers la solution du problème.

Les expériences présentées dans ce chapitre montrent la supériorité de notre algorithme pour traiter le problème d'appariement de graphes similaires ou non similaires.

Chapitre III

Application de l'algorithme d'appariement de graphes proposé pour l'indexation spatiale d'images

1. Introduction

Plusieurs travaux ont été effectués dans le passé, utilisant les graphes pour la modélisation et l'indexation de l'image. Ces travaux ont touché plusieurs domaines tels que la reconnaissance des caractères[19],la reconnaissance des objets 3D [13],la détection des routes à partir des images radar [12], l'analyse des images IRM et la reconnaissance des tumeurs [3] la reconnaissance des composantes d'un plan d'architecte [25], la reconnaissance des visages [14], etc. Cette grande diversité d'applications montre le potentiel des graphes pour la modélisation et la recherche dans les bases de données. Cependant, aucun système commercial ou éducatif de grande envergure utilisant les graphes n'a encore vu le jour. La principale raison selon nous est le manque de vulgarisation des algorithmes de recherche et de manipulation des graphes. Le mariage entre l'indexation et les outils de modélisation et de manipulation des graphes est un domaine en pleine effervescence.

Dans ce qui suit, afin de valider l'algorithme expérimentalement, nous avons construit une base de données d'images synthétiques. Des expérimentations ont été effectuées pour comparer les images.

Nous proposons en dernier lieu, une application de recherche d'images par le contenu pour les lettrines.

Dans la section suivante nous allons décrire l'architecture générale de l'application.

2. Recherche d'images synthétiques

Dans cette section, nous allons étendre notre évaluation à des données formant des images synthétiques. Le but de cette évaluation est d'une part de faire un pas vers le cas réel (image du monde réel) et d'autre part évaluer et étendre le champ d'application de l'algorithme d'appariement de graphes déjà décrit dans le chapitre 2 à plusieurs étiquettes par sommet ou arête. Cette extension, permettra à l'utilisateur d'interroger le moteur de recherche selon plusieurs critères en combinant les différentes caractéristiques des objets et des relations entre objets. Dans cette section, nous allons présenter une application de recherche d'images synthétiques par le contenu. La base de données comporte des images synthétiques générées manuellement. Les détails de la construction de la base de données, la modélisation des

images par des graphes ainsi que les résultats expérimentaux de la recherche d'images par le contenu seront exposés dans les sous sections qui suivent.

2.1. Construction de la base de données

Notre tâche en premier lieu est donc de construire une base d'images synthétiques manuellement, pour cela nous avons adopté les hypothèses suivantes.

1. Chaque image comportera au moins 2 objets et au plus 5 objets.
2. Un objet n'est autre qu'une forme géométrique régulière : carré, triangle, ellipse, losange ou un cercle (voir figure 15).
3. La forme, la taille et la position relative sont les caractéristiques de base d'un objet.

Pour que notre application fonctionne correctement, on ne doit pas superposer deux objets d'une même image. La construction d'une telle base de données d'images vise premièrement à bien distinguer la ou les images qui sont visuellement très proches et deuxièmement à faciliter la modélisation des images par des graphes, ce qui constitue le sujet de la sous-section suivante.

En plus, le nombre d'images dans la base de données a été fixé préalablement à 200, mais nous avons conçu la base de données d'une façon à pouvoir ajouter facilement d'autres images à la base de connaissances.

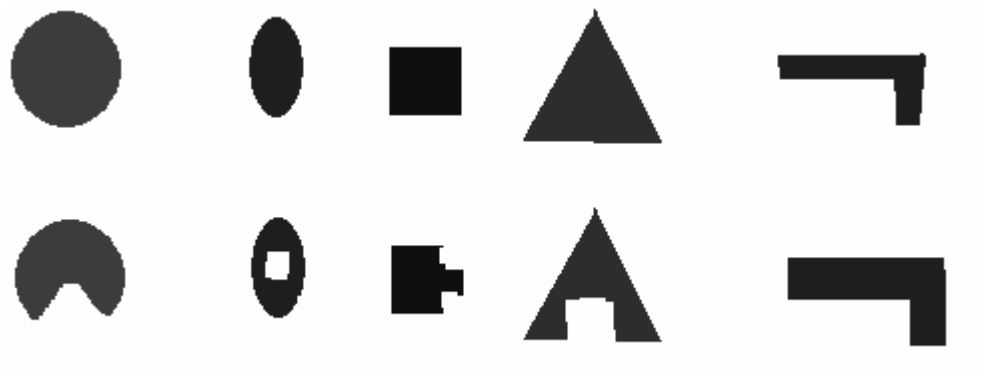


Figure 15. *Exemple d'objets utilisés pour la construction des images synthétiques.*

2.2. Modélisation des images par des graphes

L'étape de la modélisation des images par des graphes est une des plus importantes dans un processus de recherche d'images par le contenu. La première tâche consiste à définir les caractéristiques pertinentes à extraire de l'image. Une deuxième tâche consiste à définir la

structure du graphe modélisant l'image en question. L'image telle que décrite dans la section précédente est une collection d'objets réguliers non superposés. Chaque objet dans l'image est décrit par la forme et la taille. La structure du graphe modélisant une image permettra d'obtenir une représentation fidèle de l'image en question. Chaque sommet dans un graphe représentera un objet dans une image. Chaque arête reliant deux sommets dans un graphe représentera une relation entre deux objets dans une image. Un sommet est décrit par deux étiquettes représentant respectivement la forme et la taille de l'objet. Une arête est décrite par deux étiquettes représentant respectivement la distance et l'angle entre deux objets de l'image. Nous allons décrire dans la section suivante les différentes étiquettes utilisées pour la construction de graphes.

2.3. Description des attributs choisis dans la modélisation des graphes.

2.3.1. Distance

La distance entre les deux régions qu'on va utiliser est simplement la distance entre les deux centres de gravité des deux régions.

Ainsi en supposant dans la figure 16 que (Xa,Ya) et (Xb,Yb) sont les deux centres de gravité respectivement de l'objet a et b ,la distance entre ces deux régions est donnée par l'équation suivante :

$$d(a,b) = \sqrt{(Xa - Xb)^2 + (Ya - Yb)^2} \quad (11)$$

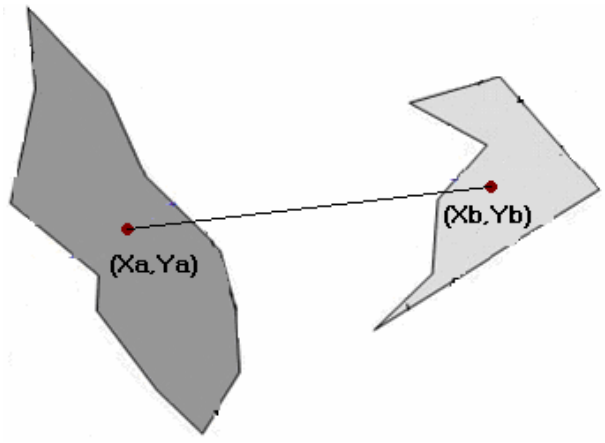


Figure 16. *Distance entre deux objets.*

La normalisation de la distance est effectuée lors du calcul de l'ecag.

2.3.2. Forme

Pour définir la forme de l'objet, plusieurs choix déjà décrits dans le chapitre 1 sont étudiés.

On va utiliser dans ce travail une description de forme grossière. En effet l'utilisation d'une description très fine peut donner parfois des résultats bizarres vu que les formes qu'on va comparer sont générées par un algorithme de segmentation. On a choisi d'utiliser les trois premiers moments de Hu pour la description des formes [20].

2.3.3. Angle

L'angle obtenu à partir de deux objets qu'on va utiliser est l'angle formé par l'axe horizontal et la droite qui passe par les deux centre de gravité des deux objets(Voir figure 16). D'après la figure 17, on déduit que l'angle le plus proche de l'horizontal est défini par la formule suivante : $\alpha = \text{arc}\,tg\left(\frac{d}{D}\right)$ (12)

Avec d et D respectivement les distances formées par les projections verticale et horizontale des deux centres de gravité des deux objets.

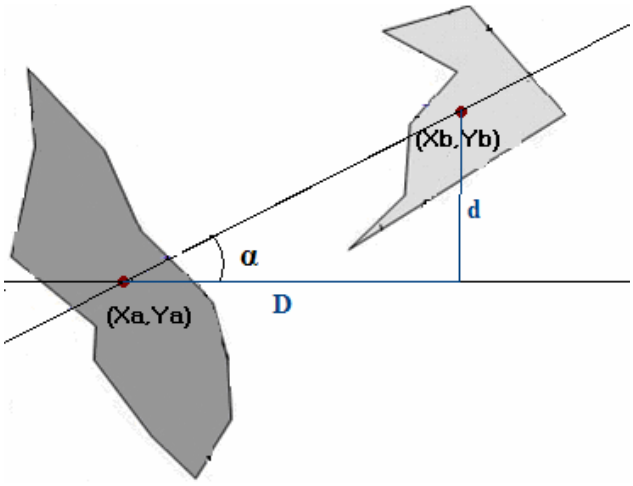


Figure 17. *Angle relatif entre deux objets.*

L'angle α donné par la formule 12 appartient à l'intervalle $\left[0, \frac{\pi}{2}\right]$

Nous remarquons dans les deux exemples de la figure 18 que l'angle entre les deux régions calculé à partir de la formule 12 est le même, bien que la disposition des deux objets dans chaque figure soit différente.

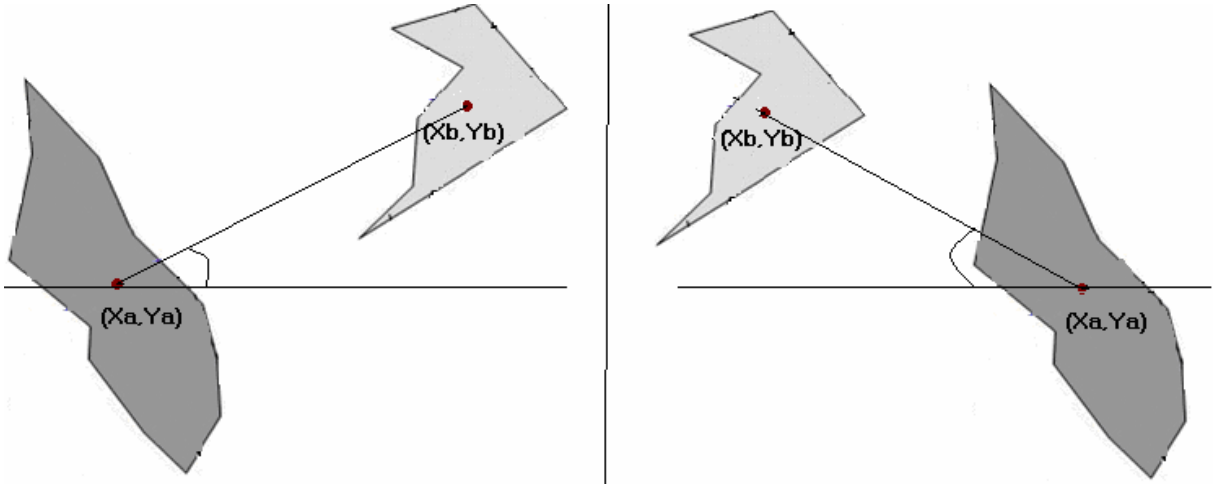


Figure 18. *Ambiguïté au niveau de la définition de l'angle relatif.*

Pour distinguer ces deux cas, on va effectuer un petit test entre les deux centres de gravité des deux objets.

L'angle α est alors défini de la manière suivante :

Si $Xa < Xb$ et $Ya < Yb$ alors $\alpha = \text{arc}\,tg\left(\frac{d}{D}\right)$, sinon, $\alpha = \pi - \text{arc}\,tg\left(\frac{d}{D}\right)$. (13)

Avec cette méthode, l'angle calculé entre les deux objets appartient à l'intervalle $[0, \pi]$.

Dans le cadre de ce projet de fin d'études le critère d'invariance par rotation dans la recherche des images par le contenu n'est pas imposé.

Mais si ce critère était imposé nous pourrions utiliser l'angle formé par les deux axes majeurs des deux objets :

Supposons que x_{ab} et x_{cd} représente les deux axes majeurs des deux objets à étudier.

L'angle α est alors défini par la formule suivante :

$$\alpha_{ab,cd} = arccos\left[\frac{x_{ab}.x_{cd}}{|x_{ab}|.|x_{cd}|}\right] (14)$$

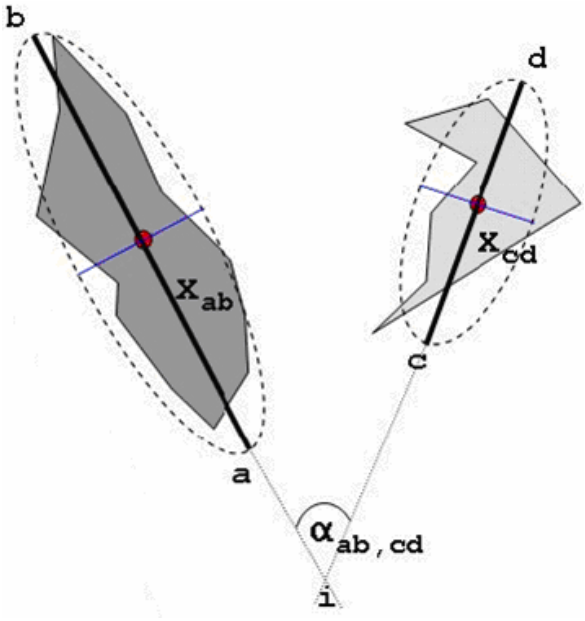


Figure 19. *Autre définition d'angle entre deux objets.*

2.3.4. Taille

La taille d'une région est définie par son nombre de pixels.

La normalisation de la taille est effectuée lors de l'étape de l'appariement de graphes.

3. Expérimentation et discussion.

L'algorithme d'appariement de graphes décrit dans le chapitre 2 permet de rechercher l'isomorphisme de graphes ou de sous-graphes existant entre deux graphes. L'algorithme a été décrit pour des graphes comportant une étiquette par sommet et par arête. Une première tâche donc consiste à adapter l'algorithme à plusieurs étiquettes par sommet ou par arête. D'autre part, nous allons utiliser dans l'algorithme d'appariement de graphes la distance Euclidienne pour mesurer la similarité entre deux étiquettes, ce qui induit une similarité globale entre les deux graphes.

L'erreur globale ou l'erreur de correction notée *ecag* est composée de l'erreur engendrée par l'appariement des sommets noté $ecag_s$ et de l'erreur engendrée par l'appariement des arêtes noté $ecag_a$.

$$ecag = \sum_{i=1}^{|v|} ecag_s + \sum_{i=1}^{|v|} ecag_a \quad (15)$$

Avec $|v|$ la taille du plus petit graphe.

D'une part, l'appariement des sommets consiste à mesurer le degré de similarité ou la distance entre les valeurs des étiquettes forme et taille, notées respectivement $ecag_{sf}$ et $ecag_{st}$.

Nous avons introduit pour chaque terme de cette dernière équation un paramètre pour contrôler son usage lors de la définition de la requête.

$$ecag_s = \alpha ecag_{sf} + \beta ecag_{st} \quad (16)$$

Ainsi l'erreur engendrée par la forme $ecag_{sf}$ prend la distance euclidienne entre les trois moments de Hu des trois moments de chaque image qui est décrite de la manière suivante :

$$ecag_{sf} = \sqrt{(M_1 - M'_1)^2 + (M_2 - M'_2)^2 + (M_3 - M'_3)^2} \quad (17)$$

Où M_1, M_2, M_3 et M'_1, M'_2, M'_3 sont respectivement les trois premiers moments de Hu du premier et du deuxième objets mis en correspondance.

L'erreur engendrée par la taille $ecag_{st}$ est décrite de la manière suivante

$$ecag_{st} = \frac{|T_1 - T_2|}{|T_1 + T_2|} \quad (18)$$

Où T1 et T2 sont les tailles respectivement du premier et du deuxième objets mis en correspondance.

D'autre part, l'appariement des arêtes noté $ecag_a$ consiste à mesurer le degré de similarité ou la distance entre les valeurs des étiquettes : distance et angle relatifs entre deux objets, appelés respectivement $ecag_{ad}$ et $ecag_{aa}$.

$$ecag_a = \gamma ecag_{ad} + \mu ecag_{aa} \quad (19)$$

Avec l'erreur engendrée par l'angle entre les deux arêtes $ecag_{aa}$ prend la distance euclidienne entre les deux valeurs des deux angles.

L'erreur engendrée par la distance $ecag_{ad}$ est décrite de la manière suivante

$$ecag_{ad} = \frac{|D_1 - D_2|}{|D_1 + D_2|} \quad (20)$$

Où D_1 et D_2 sont respectivement la distance entre deux objets de la même image concernant la première et la deuxième image.

Les différents paramètres introduits dans les équations 16 et 19 offrent à l'usager une grande flexibilité pour décrire sa requête selon ses besoins. Cette flexibilité permet à l'usager d'introduire une ou plusieurs caractéristiques et un degré d'importance pour chaque caractéristique.

3.1. Recherche d'images selon la forme

Dans cette première expérience, la recherche est effectuée selon la forme des objets. On cherche les images dans la base de données qui contiennent un carré, une ellipse, un cercle…

La valeur attribuée au paramètre α est 1, tous les autres paramètres sont nuls.

La figure 20 illustre l'image requête et le résultat de la recherche. Les quatre images résultats sont semblables à l'image requête. L'erreur relative à ces images est presque nulle.

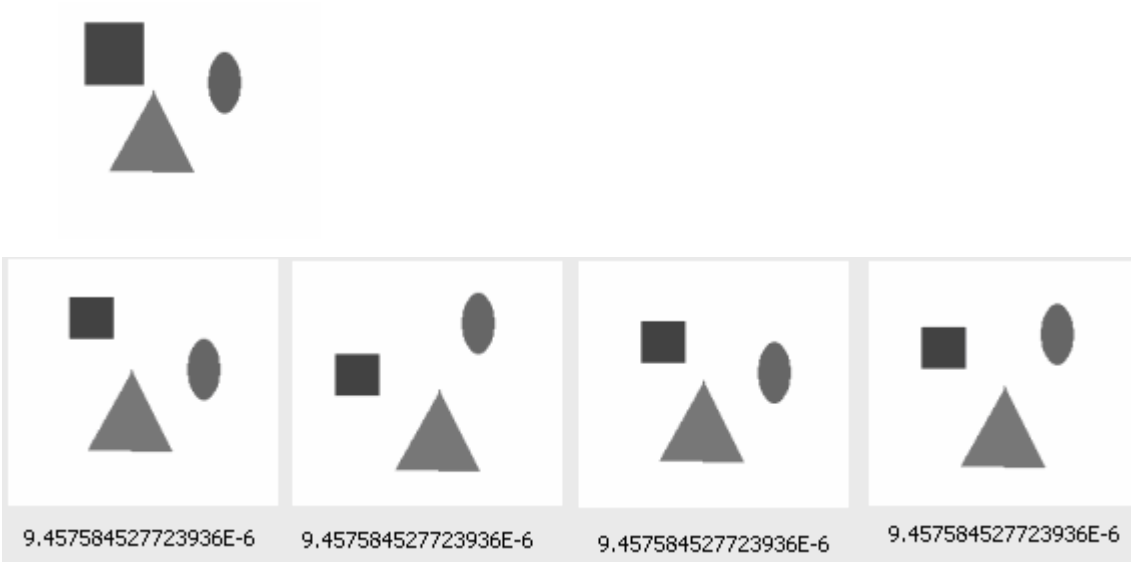


Figure 20. *Recherche d'images selon la forme ; l'image requête et les quatre images semblable.*

3.2 Recherche d'images selon la taille

Dans cette deuxième expérience, la recherche est effectuée selon la taille des objets. On cherche des images de la base de données qui ont trois objets. La valeur attribuée au paramètre β est 1, tous les autres paramètres sont nuls. L'image requête illustrée par la première image de la figure 21 comporte trois grands objets. Les quatre images résultats présentent des objets ayant la même grosseur que les objets dans l'image requête.

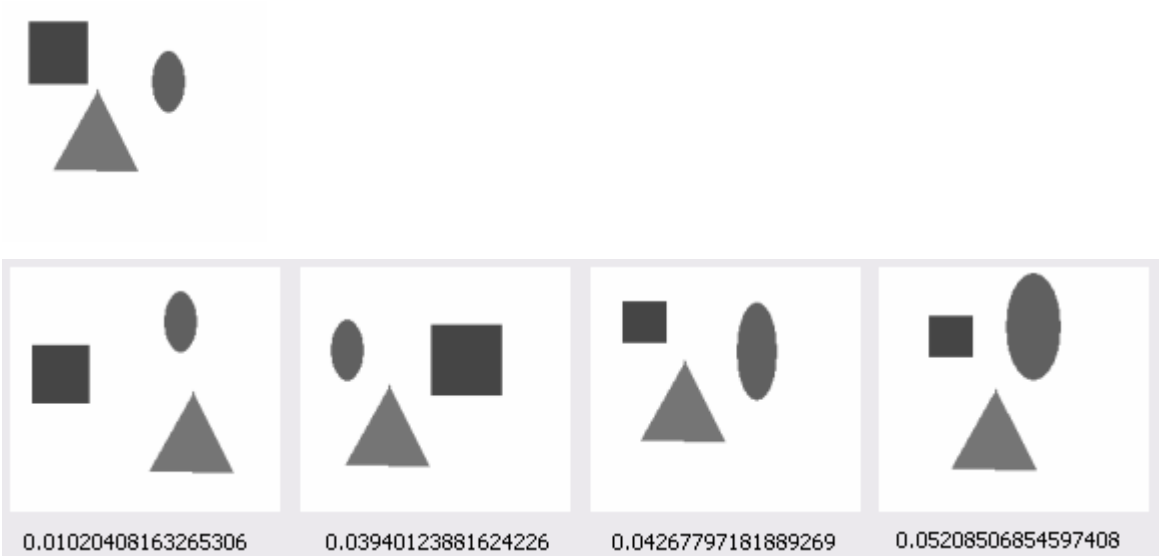


Figure 21. *Recherche d'images selon la taille ; l'image requête et les quatre images semblables.*

3.3. Recherche d'images selon la distance

Dans cette troisième expérience, la recherche est effectuée selon la distance des objets. On cherche (les images de la base de données qui ont trois objets dont les distances sont les plus proches à l'image requête).

La valeur attribuée au paramètre γ est 1, tous les autres paramètres sont nuls. Les quatre images résultats présentent une ressemblance de disposition des objets. Les images résultats ne partagent pas forcément la même forme des objets.

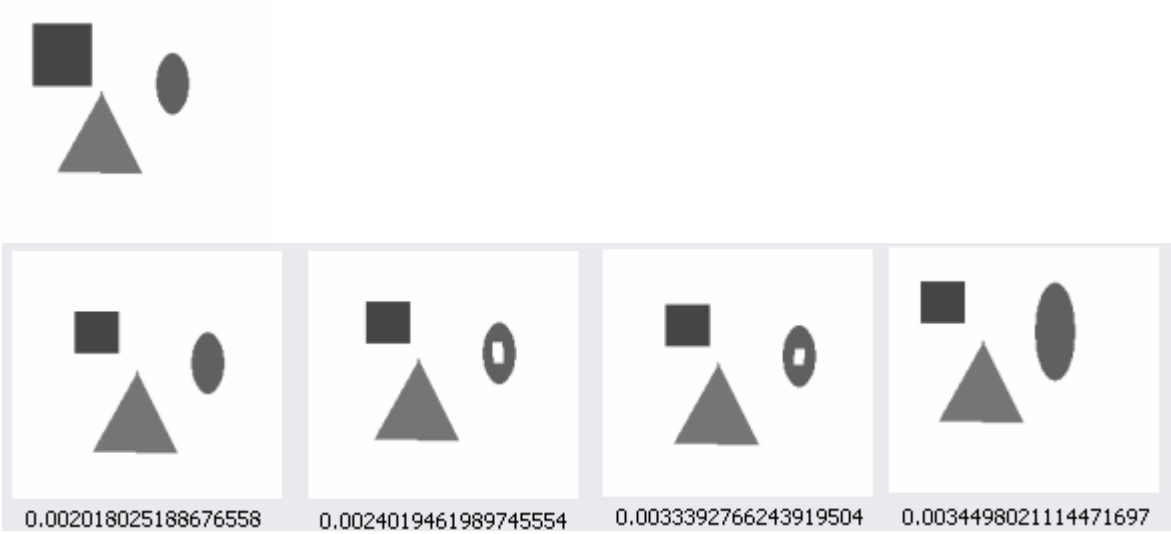


Figure 22. *Recherche d'images selon la distance relative ; l'image requête et les quatre images semblables.*

3.4. Recherche d'images selon l'angle relatif

Dans cette quatrième expérience, la recherche est effectuée selon la position relative des objets. On cherche (les images de la base de données qui ont trois objets dont les positions sont les plus proches à l'image requête).

La valeur attribuée au paramètre μ est 1, tous les autres paramètres sont nuls. Les quatre images résultats présentent la même disposition des objets. Les images résultats ne partagent pas forcément la même forme des objets.

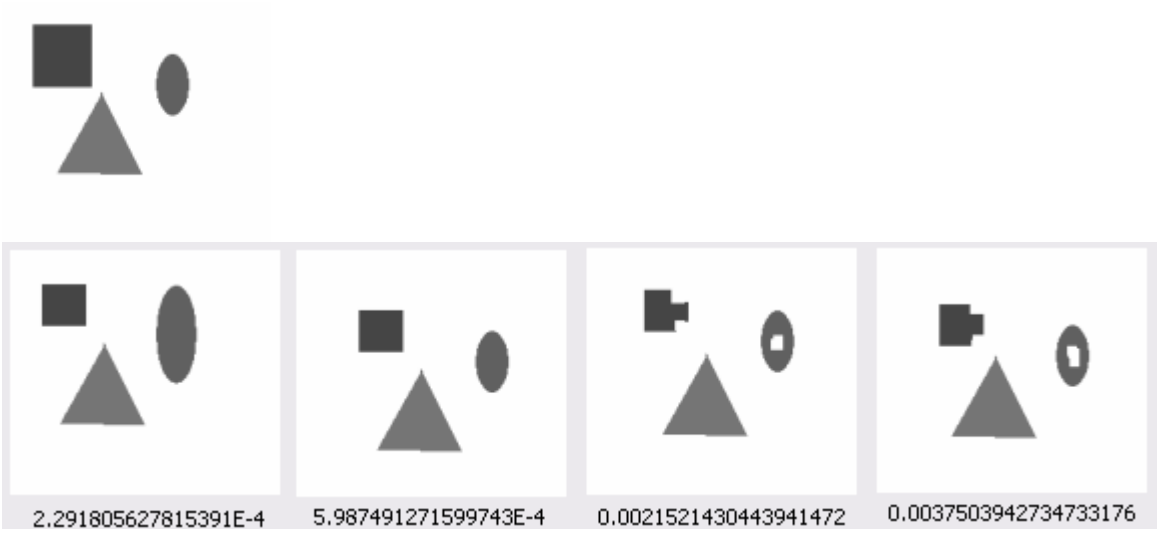


Figure 23. *Recherche d'images selon l'angle relatif; l'image requête et les quatre images semblables.*

3.5. Recherche d'images selon la forme, la taille, l'angle et la distance

Dans cette cinquième expérience, la recherche est effectuée selon la forme, la taille, l'angle et la distance. On cherche des images de la base de données qui ont trois objets et ayant la même position relative, forme et taille que l'image requête. Les paramètres α, β, μ et γ sont tous initialisés à un. Les quatre images résultats présentent la même disposition des objets ainsi que la même forme et la même taille d'objets que l'image requête. Il faut noter qu'à partir d'un certain rang des images résultats, les images ont tendance à perdre de précision concernant ou bien la position relative on bien la forme.

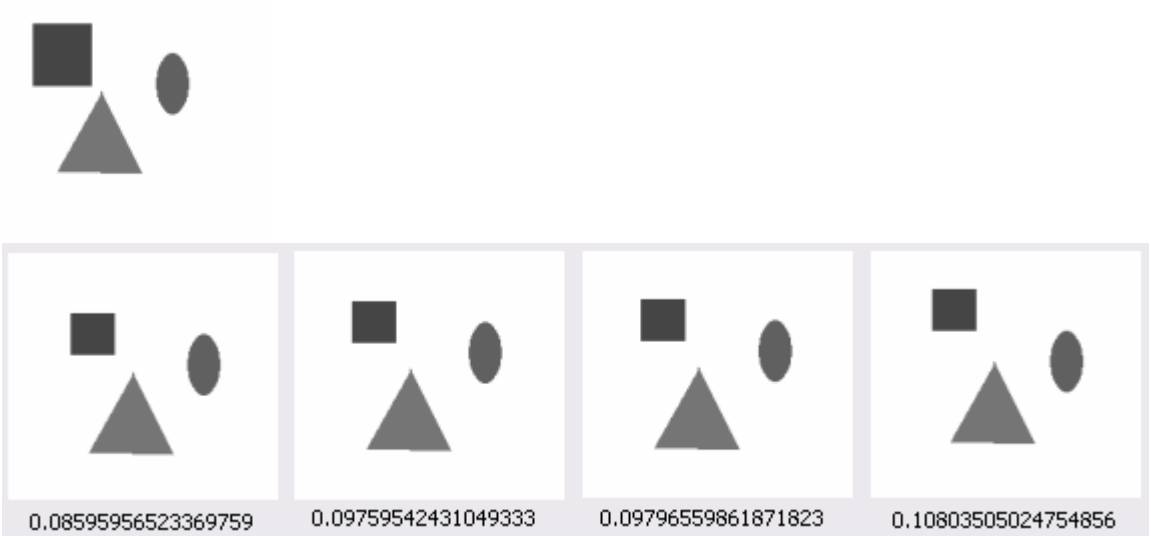


Figure 24. *Recherche d'image selon la forme, la taille, l'angle et la distance ; l'image requête et les quatre images semblables.*

4. Evaluation du moteur de recherche d'image synthétique

4.1. Mesure de performance

Pour évaluer les différents algorithmes de recherche d'image par le contenu, nous avons besoin d'une mesure de performance. On retrouve plusieurs types de mesure de performance telle que le PWH "Percentage of Weighted Hits», le RPP "Recall and Precision Pair" et le PSR "Percentage of Similarity Ranking".

Dans notre travail nous allons utiliser le RPP, on effet c'est la mesure de performance la plus citée et testée dans la littérature

La performance de la recherche est évaluée en utilisant la précision et le rappel. La précision P est définie comme étant le ratio du nombre des images pertinentes recherchées r sur le nombre total d'images trouvées n, elle mesure donc la capacité du système à retrouver les images pertinentes. Le rappel R est défini comme étant le nombre des images pertinentes recherchées sur le nombre total m des images pertinentes dans toute la base de données, il mesure donc la capacité du système à ne retrouver que les images pertinentes. La précision P et le rappel R sont données par :

$$P=\frac{r}{n} \qquad \text{et} \qquad R=\frac{r}{m}$$

Pour mesurer ces paramètres, nous supposons que la base d'images est constituée de classes d'images disjointes dont nous connaissons les cardinaux respectifs m. Toutes les images de la base sont prises successivement en tant qu'images requêtes et nous observons les images retournées jusqu'à un rang *n* variable. Pour le paramètre rappel, nous traçons les courbes $R=f(n)$. Plus vite cette courbe tendra vers 1 et plus le système sera performant. Pour le paramètre précision, Nous traçons les courbes $P=f(n)$. Plus le système, est précis et plus la courbe P décroît lentement.

4.2. Résultats faisant intervenir la forme, la taille, l'angle et la distance.

Nous avons réalisé notre expérimentation sur une base d'image synthétique en considérant 5 classes composées chacune de 30 images prises d'une façon aléatoire de la base.

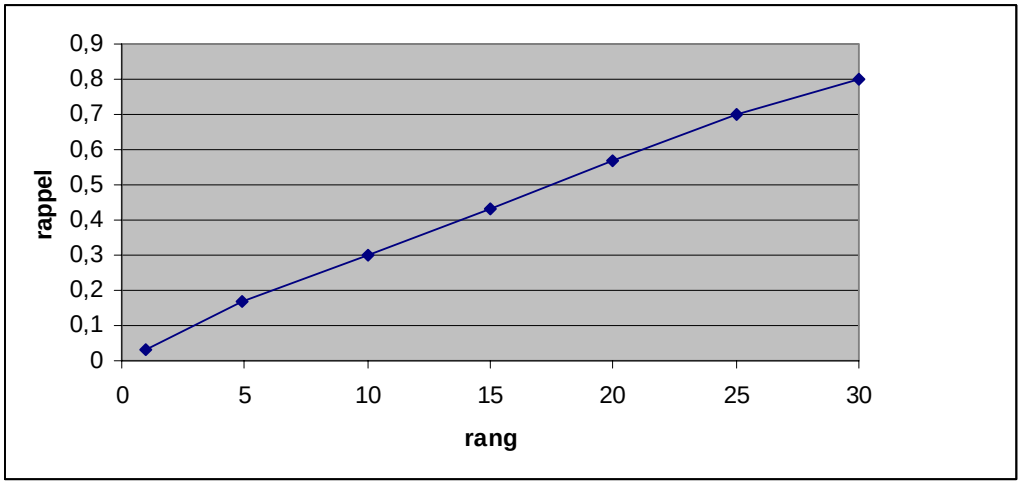


Figure 25. *Courbe de rappel faisant intervenir la forme, la taille, l'angle et la distance*

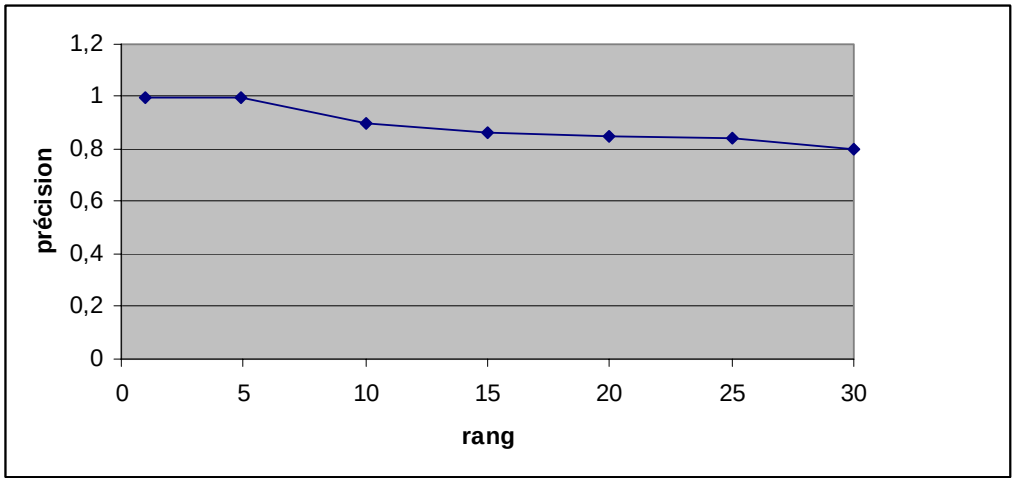


Figure 26. *Courbe de précision faisant intervenir la forme, la taille, l'angle et la distance*

La courbe de rappel de la figure 25 tend rapidement vers 1 et la courbe de précision de la figure 26 décroit lentement. Les deux courbes de rappel et de précision montrent de très bon résultat au niveau de la performance de notre moteur de recherche d'image synthétique.

5. Recherche de lettrines par des graphes

Dans la section précédente nous avons présenté une application permettant de modéliser, de rechercher et d'organiser une base d'images synthétiques à l'aide des graphes. Dans cette section, nous allons présenter une deuxième application permettant de modéliser et de rechercher des images réelles (lettrines).

L'objectif de la recherche de lettrines est de permettre les utilisateurs de lancer une lettrine requête et de trouver les lettrines les plus similaires à la requête à partir de la base de données. Dans ce contexte, nous sommes intéressés seulement par la recherche de lettrines par le contenu. Ce qui veut dire que la correspondance d'une requête et d'une lettrine dans la base de données n'est pas liée à la lettre de la lettrine. En effet la correspondance en utilisant les lettres des lettrines est relativement simple dans ce contexte.

Les détails de la modélisation des lettrines par des graphes ainsi que la recherche de lettrines par le contenu seront exposés dans les sous sections qui suivent.

5.1. Architecture de l'application de recherche de lettrines par le contenu

L'objectif de cette section est de construire un système de recherche de lettrines par le contenu efficace, faisant appel à la modélisation structurelle des images et l'algorithme d'appariement de graphes développé dans ce projet de fin d'études.

Notre système de reconnaissance de lettrines par le contenu décrit dans cette section comporte trois phases (Figure 27).

- Phase préliminaire (offline) : segmentation des lettrines de la base de données et construction des graphes modélisant ces lettrines.
- Phase en ligne : segmentation de la lettrine requête et construction de ces graphes pour obtenir la représentation de cette lettrine.
- Phase de recherche : rechercher des lettrines proches de la lettrine requête en comparant les ecags calculés à partir des graphes.

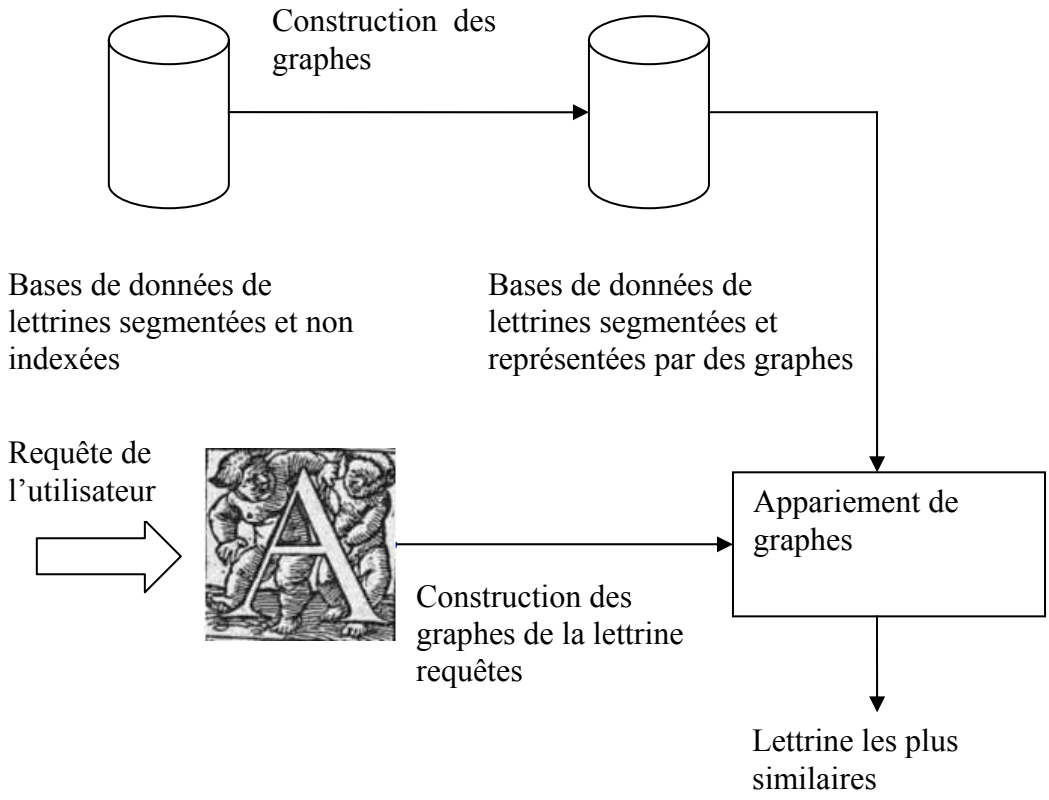


Figure 27. Synoptique du *système de recherche d'images par le contenu*

La première phase (offline) de notre système est divisée en deux étapes. La première étape consiste à segmenter chaque lettrine de la base de données. La deuxième étape consiste à la construction des graphes des images obtenues après l'étape de segmentation. A la fin de cette phase, la lettrine est modélisée par trois graphes ou chaque graphe représente l'image de chaque couche segmentée (voir section suivante).

La deuxième phase consiste à construire les trois graphes qui représentent les trois images obtenues après l'étape de segmentation de la lettrine requête.

La dernière phase est consacrée à la recherche de lettrines similaires à la lettrine requête dans la base de données d'images en faisant appel à l'algorithme d'appariement de graphes. Dans ce qui suit, nous allons présenter un aperçu sur l'étape de segmentation, la modélisation des lettrines par les graphes ainsi que les résultats expérimentaux de la recherche d'images de lettrines par le contenu.

5.2. Segmentation des lettrines

Un algorithme spécifique pour la segmentation des lettrines a été mis en place dans le laboratoire L3I.

Cet algorithme permet de segmenter la lettrine en différentes couches d'informations. Cette segmentation qui est basée sur la technique « first top-down » nous donne trois couches d'informations de l'image originale (zones texturées, zones homogènes, les contours) (figure 28).

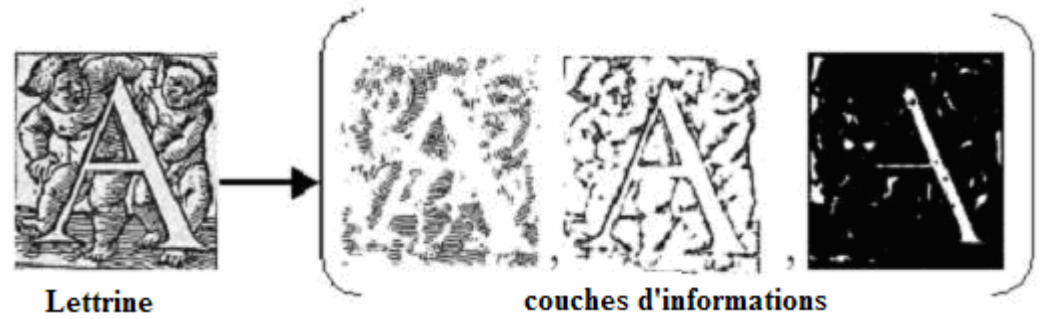


Figure 28. *Segmentation de la lettrine en trois couches d'informations (zones texturées, zones homogènes, les contours)*

Afin d'adapter cette segmentation à notre algorithme d'appariement il est nécessaire de bien distinguer les régions des couches d'informations pour cela une étape de détection et de fusion des régions est nécessaire. Le but de cette étape est de déterminer les régions connexes qui seront utilisées lors de la modélisation et de la recherche de lettrines.

Après cette étape chaque couche d'information contient entre deux et douze objets (figure 29).

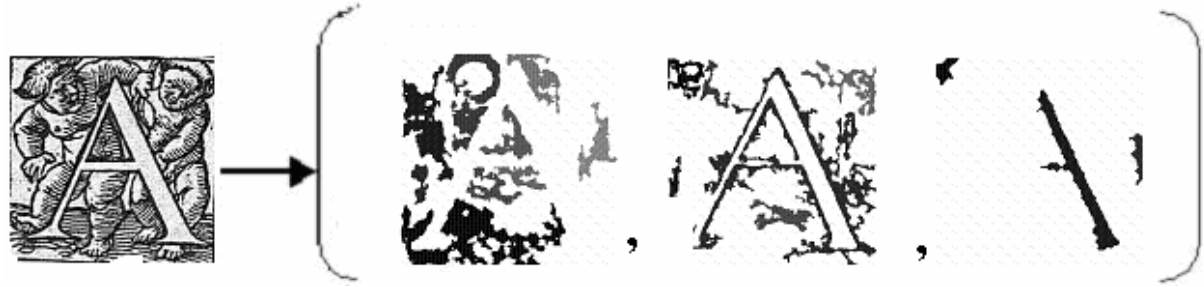


Figure 29. *Segmentation de la lettrine en trois couches d'informations libellées (zones texturées, zones homogènes, les contours)*

5.3. Modélisation des lettrines par des graphes

La structure de graphe utilisée est la même que celle utilisée dans la modélisation des images synthétiques. Dans cette application on va modéliser par des graphes les images de lettrines segmentées au lieu des images synthétiques.

Chaque sommet dans un graphe représentera un objet dans une image. Chaque arête reliant deux sommets dans un graphe représentera une relation entre deux objets dans une image. Un sommet est décrit par deux étiquettes représentant respectivement la forme et la taille de l'objet. Une arête est décrite par deux étiquettes représentant respectivement la distance et l'angle entre deux objets de l'image (voir figure 30 et 31).

La seule différence est que pour l'application des images synthétiques chaque image est représentée par un seul graphe. Par contre dans l'application des lettrines chaque image de lettrines lui correspond trois graphes. Chaque graphe modélise une couche d'informations de la lettrine.

Figure 30. *Image de la lettrine et son image segmentée (couches d'informations zones homogène)*

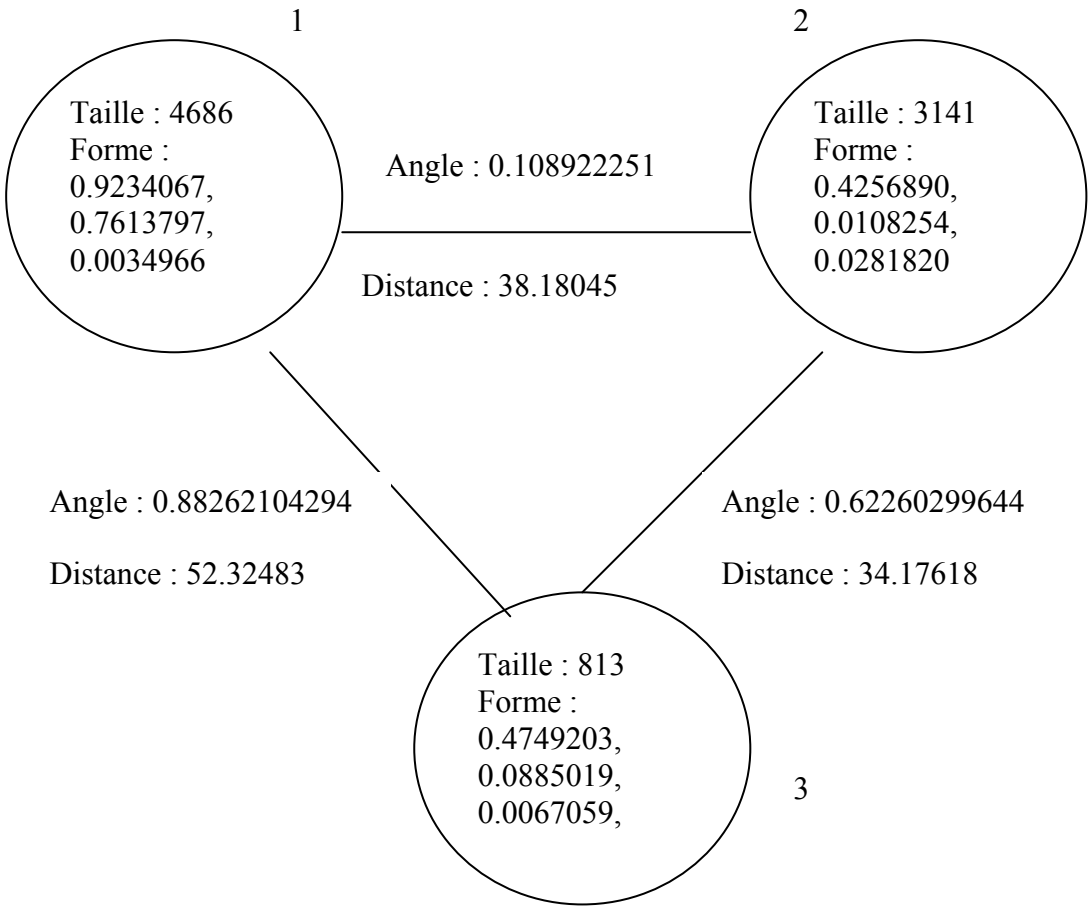


Figure 31. *Le graphe correspondant à la lettrine segmentée (couches d'informations zones homogène) de la figure 30.*

5.4. Distance entre deux lettrines.

La distance entre deux lettrines L_1, L_2 est définie par :

$D(L_1, L_2) = ecag_1 + ecag_2 + ecag_3$

Avec $ecag_1$ l'erreur globale ou l'erreur de correction engendrée par l'appariement des deux graphes modélisant respectivement la première couche d'informations de L_1 et la première couche d'informations de L_2, $ecag_2$ l'erreur globale ou l'erreur de correction engendrée par l'appariement des deux graphes modélisant respectivement la deuxième couche d'informations de L_1 et la deuxième couche d'informations de L_2, et $ecag_3$ l'erreur globale ou l'erreur de correction engendrée par l'appariement des deux graphes modélisant respectivement la troisième couche d'informations de L_1 et la troisième couche d'informations de L_2.

Notons que le calcul de $ecag_1, ecag_2$ et $ecag_3$ s'effectue de la même façon que le calcul de l'ecag défini dans l'application de recherche d'images synthétiques par la forme.

5.5. Expérimentation

La base d'images utilisée dans l'expérience contient 26 lettrines. Chaque lettrine a été segmentée en trois couches d'informations. Ensuite, chaque lettrine a été modélisée par trois graphes. Chaque graphe contient entre 2 et 12 sommets. Les étiquettes décrivant un sommet sont respectivement la taille et la forme de la région correspondante. L'étiquette décrivant une arête est la distance et l'angle entre les deux régions reliant les deux sommets reliés par cette arête. La figure 33 illustre le résultat de la recherche d'une lettrine requête illustrée par la figure 32. La première et la deuxième image résultat sont très semblables à l'image requête.

Figure 32. *La lettrine requête.*

Figure 33. *Les quatre lettrines semblables à la lettrine de la figure 27.*

Conclusion générale et perspectives

Ce projet de fin d'études s'inscrit dans le cadre général de l'indexation spatiale. Il s'intéresse plus particulièrement à la modélisation des images par les graphes. Un graphe est un ensemble de sommets et d'arêtes où chaque sommet ou arête est caractérisé par un ensemble d'étiquettes. Deux axes de recherche ont formé l'épine dorsale de ce travail : l'appariement de graphes et le développement d'un système de recherche de lettrines par le contenu.

L'appariement de graphes est un problème de mise en correspondance des sommets de deux graphes qui tient compte des mises en correspondance des arêtes. C'est un problème NP-complet. Nous avons proposé un algorithme pour rechercher le meilleur appariement entre deux graphes. Tous les algorithmes d'appariement de graphes partagent le même problème d'explosion combinatoire de l'espace de recherche. Pour atténuer cette complexité, des solutions polynomiales ont été développées. Quant à l'optimalité, aucune garantie n'est acquise. Nous avons tenté de trouver une solution au problème permettant de trouver des bons appariements sans explorer tout l'espace de recherche.

La recherche d'images par le contenu est un domaine en pleine expansion. Les systèmes existants utilisent souvent l'approche globale. Notre objectif était de développer une application dans laquelle la modélisation du contenu des images est réalisée à l'aide des graphes. Cette application nécessite aussi l'algorithme d'appariement de graphes décrits dans ce projet de fin d'études. L'utilisation des graphes pour représenter le contenu d'une image présente plusieurs avantages. Un des avantages est la simplicité de définir des requêtes combinant en même temps le contenu des objets à rechercher et les relations des objets.

Les perspectives ouvertes par notre travail sont nombreuses. D'une manière générale, l'approche spatiale, en particulier les graphes, est un axe de recherche ouvert, qui mérite une attention plus approfondie.

La première perspective concerne l'algorithme d'appariement de graphes proposé. L'algorithme recherche le meilleur appariement en comparant les sommets dans un premier temps, ensuite les arêtes. L'algorithme sélectionne les mises en correspondance suivant leur degré de similarité. On peut envisager d'inclure l'approche de relaxation probabilistique pour choisir les meilleures mises en correspondance. L'algorithme gardera la même technique de recherche en profitant du choix judicieux de l'approche Bayésienne.

Une deuxième perspective consiste à réorganiser la base de données de graphes par une classification afin de séparer l'ensemble des graphes en différent groupes. Cette classification permet d'accélérer considérablement la recherche.

Bibliographie

[1] A. Hlaoui and S. Wang.: “A new algorithm for inexact graph matching”. In Proceedings of the 16 IEEE ICPR, Quebec, Canada, August, 11-15, 2002.
[2] A. K. C. Wong, S. W. Lu and M. Rioux.: “Recognition and shape synthesis of 3-D objects based on attributed hyper graphs”. In IEEE PAMI, 11(3):279-289, 1989.
[3] BUNKE, H.: “A network based approach to exact and inexact graph matching”. In Technical Report IAM-93-021, University of Bern, Institut für Informatik und angewandte Mathematik, 1993.
[4] Bunke, H.: “Graph matching: Theoretical foundations, algorithms, and applications”. In Vision Interface 2000, Montreal, 82 – 88, 2000.
[5] Champin, P.-A. et C. Solnon. “Measuring the similarity of labeled graphs”. In 5th International Conference on Case-Based Reasoning (ICCBR 2003), pp. 80–95. Lecture Notes in Artificial Intelligence 2689-Springer-Verlag, 2003.

[6] D. Zhang, G. Lu.: “A Comparative Study of Three Region Shape Descriptors”, In DICTA2002, Janvier 2002.

[7] D. Zhang, G. Lu.: “Improving Retrieval Performance of Zernike Moment Descriptor on Affined Shapes”, In Monash University, 2001.

[8] D. Zhang: “Image Retrieval Based on Shape”. In thèse de doctorat Monash University, Mars 2002.
[9] Dorigo, M. et G. D. Caro. “The ant colony optimization meta-heuristic”. In New Ideas in Optimization, pp. 11–32. McGraw Hill, London, UK, 1999.
[10] Dorigo, M. et T. Stützle.: “Ant Colony Optimization”. MIT Press, 2004.
[11] E.G.M.Petrakis: “Image Representation, Indexing and Retrieval based on Spatial Relationships and Properties of Objects”. In thèse doctorat, Mars 1993.
[12] E.G.M.Patrakis.: “Design and evaluation of spatial similarity approaches for image retrieval”. In Image and Vision Computing 20, 59-76, 2002.
[13] E. K. Wong.: “Model matching in robot vision by subgraph isomorphism”. In Pattern Recognition, 25(3) :287-303, 1992.

[14] G. J. Lu and A. Sajjanhar.: “Region-based Shape Representation and Similarity Measure Suitable for Content-based Image Retrieval”. In Multimedia Systems, 7(2):165-174, 1999.
[15] Glover F.: “Tabu search - part I. Journal on Computing”, 190–260, 1989.

[16] H. G. Barrow and R. J. Popplestone.: “Relational descriptions in picture processing”. In Machine Intelligence, 4:377-396, 1971.
[17] H. G. Barrow and R. M.: “Burstall. Suhgraph isomorphism, matching relational structures and maximal cliques”. In Information Processing Letters, 4(4):83-84, 1976.
[18] J. Llados, E. Marti and J. J. Villanueva.: “Symbol recognition by error tolerant subgraph matching between region adjacency graphs”. In IEEE FA MI, 23(10):1137-1143, 2001.
[19] J. Rocha and T. Pavlidis.: “A shape analysis model with applications to a character recognition system”. In IEEE PAMI, 16(4) :393-404, 1994.

[20] M. K. Hu.: “Visual pattern Recognition by Moment Invariants”. In IRE Transactions on Information Theory, IT-8:179-187, 1962.
[21] Petrovic, S., G. Kendall, et Y. Yang.: “A Tabu Search Approach for Graph-Structured Case Retrieval”. In STAIRS 2002, pp. 55–64,2002.
[22] S. Sorlin, O. Sammoud, C. Solnon et J. Jolion.: “ Mesurer la Similarité de Graphes”. In Atelier ECOI 2006.

[23] S. X. Liao, M. Pawlak. “Image Analysis with Moment Descriptor”, 1995.
[24] T. Leung, M. Buri, and P. Perona.: “Finding Faces in Cluttered Scenes Using Labelled Random Graph Matching”. In Proceedings of the 5 International Conference of Computer Vision, pages 637-644, 1995.
[25] Y. Wang, K. Fan and J. Horng.: “Genetic-based search for error-correcting graph isomorphism”. In IEEE SMC, 27(4):588-597, 1997.

Développement d'une méthode d'indexation spatiale pour la recherche d'images par le contenu

Ali KARRAY

الخلاصة : نظرا إلى ارتفاع عدد الصور المتواجدة على شبكة الإنترنت، كان من الضروري الحصول على وسيلة جيدة للبحث تكون سريعة الاستخراج و سهلة الخزن : وهي "أندكساسيون" عبر المحتوى، أو البحث عن طريق الصورة، والتي تعتبر جواب لهذه التساؤلات. في هذا العمل، تمت دراسة طريقة أندكساسيون مكانية جديدة معتمدة على الرسم البياني .

المفاتيح : أندكساسيون ،مطابقة الرسم البياني ،طلب عبر المثل، رسم بياني نسبي ،معيار التشابه ، الشكل.

Résumé : Vue l'augmentation du nombre d'images dans le web, il a fallu se doter de moyen de recherche performant, rapide à calculer et facile à répertorier. L'indexation par le contenu, ou recherche par l'image, vient répondre à cette interrogation. Une proposition d'une nouvelle approche d'indexation spatiale basée sur les graphes attribués a été présentée dans ce travail.

Mots clés : Indexation, Appariement de graphes, Requête par l'exemple, Graphe attribué, Mesure de similarité, Forme.

Abstract : Since the increase of the number of picture in the web, it is necessary to provide a performing mean of research fast to calculate and easy to stock. The indexing by content, or research by picture, comes to answer this question. A proposition of a new approach of spatial indexation based on attributed graphs has been presented in this work.

Key-words : Indexation, Graph matching , Request by example , Attributed graph , Similarity measurement, Shape.