Recherche d'images par noyaux sur graphes de régions

Philippe-Henri Gosselin*, Justine Lebrun* et Sylvie Philipp-Foliguet*

*ETIS CNRS
6 ave du Ponceau
95014 Cergy-Pontoise Cedex
{gosselin,lebrun,philipp}@ensea.fr

Résumé. Dans le cadre de la recherche interactive d'images dans une base de données, nous nous intéressons à des mesures de similarité d'image qui permettent d'améliorer l'apprentissage et utilisables en temps réel lors de la recherche. Les images sont représentées sous la forme de graphes d'adjacence de régions floues. Pour comparer des graphes valués nous employons des noyaux de graphes s'appuyant sur des ensembles de chaînes, extraites des graphes comparés. Nous proposons un cadre général permettant l'emploi de différents noyaux et différents types de chaînes(sans cycle, avec boucles) autorisant des appariements inexacts. Nous avons effectué des comparaisons sur deux bases issues de Columbia et Caltech et montré que des chaînes de très faible dimension (longueur inférieur à 3) sont les plus efficaces pour retrouver des classes d'objets.

1 Introduction

Le problème de la comparaison de graphes est un sujet qui a été largement étudié dans la littérature depuis plusieurs décennies. S'il existe des algorithmes pour la recherche d'isomorphisme entre deux graphes, c'est-à-dire dans le cas où les deux graphes ont la même structure, même nombre de nœuds et même nombre d'arêtes, le cas plus général de comparaison entre deux graphes de tailles différentes est un problème NP-complet. Le problème est encore plus difficile lorsque les graphes sont valués et que l'on recherche une mesure de similarité entre graphes, afin de pouvoir les ordonner, les classer, etc.

On est confronté à ce problème dans certaines approches de la reconnaissance des formes où on cherche à construire des classes d'objets représentés par des ensembles structurés de régions, lignes, points, etc. Une des problématiques de la recherche d'image par le contenu est de retrouver dans une base, les images contenant un objet particulier ou un type d'objet, d'animal ou de personne, pouvant prendre des aspects très variables dans des environnements eux aussi variables. Les signatures globales ne permettent pas toujours de résoudre ce problème et les approches par points d'intérêt ne sont pas bien adaptées aux changements d'aspect d'un animal ou d'une personne, selon la prise de vue. Une approche prometteuse semble donc être de représenter un objet par un ensemble de régions adjacentes valuées à la fois par des caractéristiques intrinsèques de couleur, texture et forme, mais aussi par leurs dispositions relatives (*cf.* Philipp-Foliguet et Gony (2006)). Le graphe d'adjacence de régions constitue donc la structure

adaptée pour représenter des objets dans leur infinie variabilité. Cependant l'obtention des régions est un problème extrêmement difficile, qui ne possède en aucun cas une solution unique, car dépendant du niveau de détail souhaité, et qui est peu robuste aux changements d'éclairage, de résolution et d'aspect de l'objet. Le nombre et les caractéristiques des régions sont donc très variables d'une image à l'autre pour représenter un même objet.

Des approches récentes ont proposé de considérer les graphes comme des ensembles de chaînes (Kashima et Tsuboi (2004)) et les similarités employant des noyaux sur des graphes se ramènent alors à des noyaux sur des chaînes. Nous avons adopté cette approche par noyau de chaînes et proposons de comparer différents types de chaînes et différentes fonctions noyau sur ces chaînes.

Pour effectuer l'appariement entre graphes ou sous-graphes d'images en un temps compatible avec une utilisation en temps réel, nous avons opté pour l'algorithme de "branch and bound". La classification ou la fouille d'une base à partir d'un ou plusieurs exemples se fait ensuite par apprentissage interactif à l'aide de SVM (Support Vector Machine) pour la classification et de techniques d'apprentissage actif pour la sélection des images à faire annoter par l'utilisateur.

2 Mesures de similarité de graphes

Chaque image de la base est représentée par un graphe $G \in \mathcal{G}$ défini par un couple $G = (V, E)$, où V est un ensemble de sommets, et $E \subseteq V \times V$ un ensemble d'arêtes. Par exemple, lorsqu'une image est segmentée en régions, on peut construire un tel graphe en considérant que chaque sommet $v \in V$ est une région, et chaque arête $e = (v_1, v_2) \in V \times V$ représente une adjacence entre deux régions. On considère aussi les chaînes $h \in H(G)$ présentes dans le graphe, à savoir les suites $(v_1, e_1, v_2, e_2, ...)$ de sommets $v_i \in V$ reliées par des arêtes $e_i = (v_{i-1}, v_i) \in V \times V$. Différentes fonctions $H(.)$ qui à un graphe G font correspondre un ensemble de chaînes peuvent être considérées, comme celles qui renvoient les chaînes avec ou sans cycles par exemple. Dans cette partie, nous nous intéressons aux fonctions de similarité qui peuvent être utilisées avec n'importe quel ensemble de chaînes, nous abordons le choix des fonctions $H(.)$ dans la partie suivante.

Dans de nombreuses mesures de similarité $S(G, G')$ entre deux graphes $G = (V, E)$ et $G' = (V', E')$ qui ont été proposées, l'idée est généralement de trouver les meilleurs appariements entre les sommets et les arêtes. Par exemple, Sorlin et al. (2006) propose une mesure de similarité qui renvoie la valeur moyenne des meilleures similarités en fonction des appariements entre sommets et arêtes. FReBIR (*cf.* Philipp-Foliguet et Gony (2006)) calcule la valeur du meilleur appariement entre une chaîne requête et les chaînes de l'autre image. Cependant cette mesure de similarité ne respecte aucune propriété mathématique usuelle telle que la symétrie ou l'inégalité triangulaire, ce qui la rend difficilement utilisable par certains outils puissants utilisés en classification ou en *"browsing"* par exemple.

Certaines mesures de similarité s'efforcent de respecter un ensemble de contraintes mathématiques permettant une utilisation facile par les moteurs de recherche, par exemple les fonctions noyaux au sens de Mercer. Dans ce cas, la mesure de similarité que nous noterons $K(G, G')$ sera le produit scalaire $K(G, G') = \langle \Phi(G), \Phi(G') \rangle$ dans un certain espace Hilbertien $\mathcal{H}$, avec $\Phi : \mathcal{G} \rightarrow \mathcal{H}$ une fonction d'injection qui à un graphe fait correspondre un vecteur de $\mathcal{H}$.

Certaines approches de construction de fonctions noyaux s'intéressent au calcul explicite du vecteur $\Phi(G)$. Par exemple, Jurie et Triggs (2005) proposent de calculer un dictionnaire des prototypes de sommets du graphe (des points d'intérêt) les plus répandus dans la base, puis projettent les sommets sur ce dictionnaire pour former des histogrammes – histogrammes qui seront les vecteurs $\Phi(G)$ de l'espace Hilbertien. Grauman et Darell (2005) intègrent les contraintes spatiales de manière implicite via une approche pyramidale. L'inconvénient de ces méthodes basées sur des prototypes est leur faible capacité à généraliser.

Une autre technique pour construire des fonctions noyaux sur graphes à laquelle nous nous intéressons plus particulièrement dans cet article, est le calcul implicite des images dans l'espace Hilbertien via la fonction noyau. Le processus de construction repose principalement sur un ensemble de propriétés concernant les fonctions noyaux, comme le fait que la somme ou le produit de deux fonctions noyaux est encore une fonction noyau. Par exemple, dans le cas où la fonction noyau ne porte que sur les sommets du graphe, la fonction suivante est une fonction noyau sous réserve que la fonction $K_V(v, v')$ l'est aussi :

$$K_{naif}(G, G') = \sum_{v \in G, v' \in G'} K_V(v, v') \tag{1}$$

Kashima et Tsuboi (2004) ont proposé de comparer deux graphes en comparant tous les parcours possibles de ces deux graphes le long des arêtes. La fonction noyau porte alors sur des ensembles (ou sacs) de chaînes, lesquels intègrent les similarités entre sommets et entre arêtes. Ils ont défini un modèle assez général pour le calcul d'une fonction noyau sur graphe, en considérant l'ensemble des chaînes $h = (v_1, e_1, v_2, e_2, ...)$ du graphe, puis en calculant la valeur moyenne de la similarité entre les chaînes de G et G' de même longueur. Si on note $|h|$ la longueur de la chaîne h, i.e. son nombre d'arêtes, cette fonction noyau peut s'exprimer :

$$K_{Kashima}(G, G') = \sum_{h \in H(G)} \sum_{\substack{h' \in H(G') \\ |h| = |h'|}} K_C(h, h') p(h|G) p(h'|G') \tag{2}$$

avec $p(h|G)$ la probabilité de trouver la chaîne h dans le graphe G.

La fonction noyau $K_C(h, h')$ mesure la similarité entre deux chaînes :

$$K_C(h, h') = K_V(v_0, v'_0) \prod_{i=1}^{|h|} K_E(e_i, e'_i) K_V(v_i, v'_i) \tag{3}$$

Les noyaux mineurs qui interviennent dans cette équation sont K_V, noyau sur les sommets et K_E, noyau sur les arêtes. Pour K_V, nous utilisons classiquement un noyau Gaussien, qui retourne des valeurs comprises entre 0 et 1. Le noyau K_E permet de prendre en compte les similarité entre arêtes (*cf.* Suard et al. (2005)), ou plus simplement il peut prendre une valeur fixe.

Dans le contexte de graphes sur des molécules utilisé par Kashima, la similarité entre sommets est binaire, un sommet (un atome) est ou n'est pas le même qu'un autre. Cependant, dans notre contexte où la similarité entre deux sommets est continue, cette fonction a tendance à noyer dans la somme les similarités entre chaînes. Par exemple s'il existe 3 appariements (valeur de similarité a élevée) parmi 10000 appariements possibles (9997 valeurs de similarité b

faibles), alors la similarité globale vaudra $3a + 9997b$. Les 3 appariements forts ne faisant pas le poids face aux 9997 appariements faibles. Dans le contexte d'application aux images, les probabilités (connues pour des molécules) de l'équation **?** sont toutes mises à 1.

L'autre problème relatif au modèle de Kashima concerne sa complexité très élevée en terme de calculs. Afin de palier ces problèmes Suard et al. (2005) a proposé une fonction noyau aux propriétés plus intéressantes pour notre contexte. Il utilise une recherche du maximum, ce qui permet d'une part d'utiliser des algorithmes rapides, mais aussi d'avoir une meilleure discrimination :

$$K_{Suard}(G,G') = \frac{1}{2}\left(\sum_{h\in H(G)}\sum_{|h|=l}\max_{|h'|=l} K_C(h,h') + \sum_{h'\in H(G')}\sum_{|h'|=l}\max_{|h|=l} K_C(h,h')\right) \quad (4)$$

Notons que cette fonction ne respecte pas les conditions de Mercer, cependant elles ne sont violées que dans des cas très particuliers, et s'avèrent toujours vérifiées sur les bases de données que nous avons utilisées. Ce type de fonctions a aussi été utilisé par Eichhorn et Chapelle (2004) et Wallraven et al. (2003) qui en tirent les mêmes conclusions.

D'autres fonctions permettent d'augmenter encore plus la discrimination ainsi que la vitesse de calcul, avec un processus de mise en correspondance proche du moteur FReBIR (*cf.* Philipp-Foliguet et Gony (2006)), qui effectue la recherche du meilleur appariement :

$$K_{max}(G,G') = \max_{h\in H(G)} \max_{\substack{h'\in H(G')\\ |h|=|h'|}} K_C(h,h') \quad (5)$$

De même, cette fonction n'est pas une fonction noyau au sens strict, cependant en pratique elle respecte aussi les conditions sur les bases expérimentales.

3 Appariement de graphes

3.1 Algorithmes d'optimisation

Une fois définie la mesure de similarité entre deux graphes, le problème de trouver l'appariement qui maximise cette similarité demeure très complexe, surtout si on ne se limite pas aux isomorphismes entre les deux graphes. Il y a souvent un compromis à faire entre solution optimale et temps de calcul. Les algorithmes par colonie de fourmis ou par recherche taboue (*cf.* Sorlin et al. (2006)) trouvent des solutions optimales mais sont trop longs pour les utilisations "temps réel" que nous envisageons. Une autre approche très répandue utilise des arbres de recherche. Chaque noeud de cet arbre représente un couple de sommets $(v, v\prime) \in V \times V'$ candidats à l'appariement. On construit une arborescence de proche en proche à partir d'un noeud racine vide et en développant chaque noeud par les couples candidats. Les noeuds candidats sont les couples $(v, v\prime)$ compatibles avec les noeuds déjà présents dans le chemin menant de la racine au noeud courant. L'avantage de cette représentation par arborescence est que la similarité d'un chemin se calcule au fur et à mesure de la construction du chemin. Dans le cas d'une fonction de similarité qui utilise un max (comme K_{Suard} ou $K_{\max}$), l'algorithme "branch and bound" permet de trouver une solution optimale sans explorer toutes les solutions possibles. On obtient d'abord la solution la plus prometteuse qui fournit une borne inférieure

de la similarité $K(G, G\prime)$. Puis on construit les autres branches de l'arbre seulement si elles sont susceptibles d'améliorer la valeur de similarité. La solution la plus prometteuse est obtenue en explorant d'abord le chemin construit avec les noeuds dont les valeurs de similarité sont les plus grandes.

3.2 Ensembles des chaînes

Puisque nous avons choisi de mesurer la similarité entre graphes par la similarité entre ensembles de chaînes, nous décrivons ici différentes propriétés que peuvent posséder les ensembles de chaînes et qui correspondent à des configurations d'appariements.

On notera pour simplifier $h = abc...$ une chaîne du graphe G , avec $a, b, c \in V$ et $h' = a'b'c'...$ une chaîne du graphe G'avec $a', b', c' \in V'$.

La plupart de noyaux cités dans la section 2 s'appliquent sur des chaînes de même longueur. Pour pouvoir comparer des chaînes de longueurs différentes, il suffit d'autoriser les boucles : par exemple pour comparer abc et $a'b'$, on peut comparer abc et $a'a'b'$. Ainsi on ne se limite pas aux isomorphismes entre les deux graphes ce qui signifie en terme d'images, qu'on apparie trois régions d'une image avec deux régions de l'autre image. D'autres chaînes particulières sont les cycles dans lesquels les deux sommets extrémités sont identiques (exemple : $abcda$) et les chaînes Eulériennes dont chaque arête est parcourue au plus une fois ($abcdbc$ interdit).

On peut construire différents ensembles de chaînes $H(G)$ possédant certaines de ces propriétés, par exemple :

- $H_t(G)$: toutes les chaînes sans restrictions ;
- $H_{sb}(G)$: chaînes sans boucle (employées par Kashima et Tsuboi (2004)) ;
- $H_{sbc}(G)$: chaînes sans cycle ni boucle (employées par Suard et al. (2005)) ;
- $H_E(G)$: chaînes Eulériennes (employées par Philipp-Foliguet et Gony (2006)).

On note $H^h(G)$ l'ensemble des chaînes de longueur h de G et $H^{\leq h}(G)$ l'ensemble des chaînes de longueur $\leq h$ de G. Le nombre de chaînes de chacun des ensembles $H(G)$ est comparé dans la table 1 pour des longueurs de 1, 2 et 3.

	$H^h_{sb}(G)$	$H^h_{sbc}(G)$	$H^h_E(G)$
$\|h\| = 1$	20	20	20
$\|h\| = 2$	80	60	60
$\|h\| = 3$	320	120	180

TAB. 1 – *Cardinal de différents ensembles de chaînes de longueur fixe pour un graphe complet possédant 5 sommets.*

4 Expérimentations

Nous avons utilisé deux bases d'images pour nos tests : Columbia modifiée et Caltech, où l'objectif est de retrouver les images qui contiennent un objet particulier ou un objet appartenant à une catégorie. La première nous sert à tester et comprendre le comportement des

différents noyaux selon la longueur des chaînes employées. La deuxième permet de comparer sur une base réelle différents ensembles de chaînes et aussi de comparer les méthodes par graphe d'adjacence de régions à d'autres méthodes soit globales, soit employant des ensembles de point d'intérêt.

Les images sont segmentées en régions floues (*cf.* Philipp-Foliguet et Gony (2006)), ce qui permet de segmenter une base entière avec un réglage de paramètres global sur la base. Les régions sont des ensembles flous, qui se chevauchent plus ou moins (cf Fig 3). Le nombre de régions formant l'objet est très variable d'une image à l'autre, un visage par exemple peut être constitué par une seule région floue sur une image ou par 5 comme sur la Fig 3. Chaque région est représentée par un histogramme de 32 valeurs, 8 valeurs pour la couleur (chrominances de $L^\star a^\star b^\star$) et 24 valeurs pour le gradient relatif à l'orientation principale de la région (3 échelles et 8 orientations).

4.1 Protocole expérimental

Nous possédons une vérité terrain pour chacune des bases, ce qui nous permet d'une part de simuler la recherche interactive d'un objet ou d'une classe d'objets, et d'autre part d'évaluer les résultats renvoyés par le système. Muni d'un noyau sur graphe, nous entraînons un classifieur SVM dans le but de classer les images par ordre de pertinence. Les noyaux nous permettent aussi d'utiliser des techniques d'apprentissage actif qui permettent de sélectionner les meilleures images à faire annoter par l'utilisateur (*cf.* Gosselin et Cord (2006)). Nous évaluons chaque méthode par la simulation d'un grand nombre de sessions de recherche. Pour chaque session de recherche, une catégorie est choisie. Au sein de cette catégorie, une image choisie au hasard est annotée positivement, ce qui permet d'obtenir un premier classement en se basant uniquement sur la similarité. Puis, les images sélectionnées par la technique d'apprentissage actif sont annotées en fonction de la catégorie choisie au début de la session simulée. Ce premier jeu d'annotation permet d'entraîner le classifieur et ainsi d'obtenir un meilleur classement. Le processus est ensuite répété en suivant le même principe de sélection et de classification. Nous répétons la simulation de session de recherche une centaine de fois pour chaque catégorie. Ainsi, pour chaque catégorie nous pouvons mesurer la qualité moyenne du classement à chaque étape d'annotation à l'aide de la Précision Moyenne, un critère d'évaluation souvent utilisé dans les campagnes d'évaluation telle que TRECVID[1]. Puis, dans le but d'obtenir une mesure de la qualité globale du système, nous calculons la valeur moyenne des Précisions Moyennes (MAP, Mean Average Precision) sur toutes les catégories.

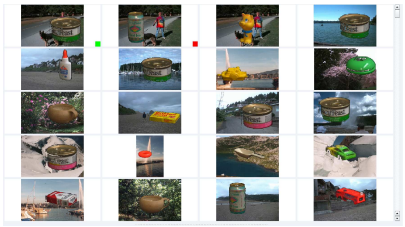
4.2 Columbia modifiée

La base Columbia[2] modifiée contient environ 600 images réparties en 50 objets de 12 vues chacun et placés sur un fond aléatoire. Ces images sont générées à partir d'images d'objets de Columbia dans lesquelles on remplace le fond par une image de paysage issue de la base ANN[3] (*cf* Fig. 1). Les graphes issus de la segmentation de la base ont entre 3 et 15 sommets, le nombre moyen de sommets est d'environ 10. Les objets sont constitués d'une à trois régions, les autres régions constituant le fond.

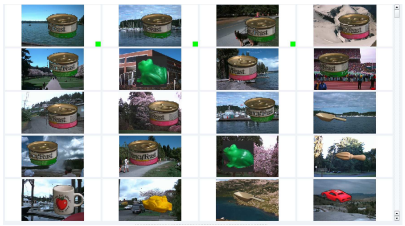
[1] http ://www-nlpir.nist.gov/projects/trecvid/

[2] http ://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[3] http ://www.cs.washington.edu.=/research/imagedatabase/

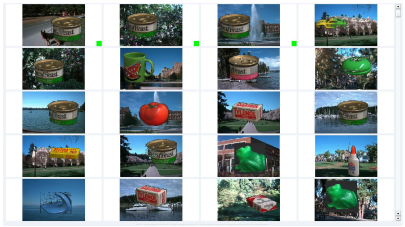


(a0) Classement initial avec K_{max}



(a1) Avec 5 annotations

(b0) Classement initial avec $K_{kashima}$



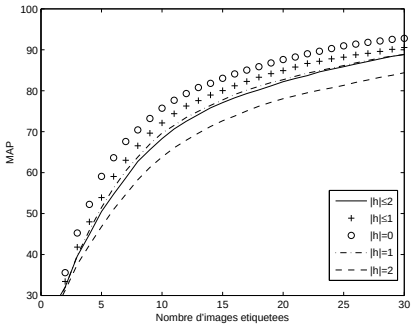
(b1) Avec 5 annotations

FIG. 1 – *Exemples de recherche sur la base Columbia modifiée. Chaque ligne présente une recherche soit avec le noyau K_{max} et l'ensemble de chaînes $H^{\leq 2}$ (a0,a1), soit avec le noyau $K_{kashima}$ et l'ensemble de chaînes H^0 (b0,b1). Sur la colonne de gauche (a0,b0) on peut voir le classement initial avec une annotation positive. Sur la colonne de droite (a1,b1) on peut voir le classement avec 3 annotations positives et 2 négatives.*

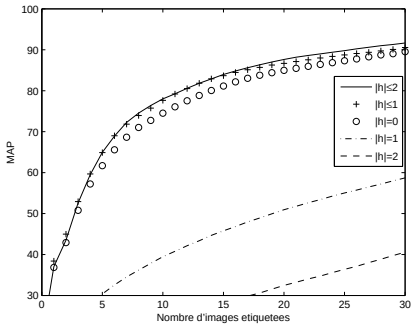
4.2.1 Comportement des noyaux en fonction de la longueur des chemins considérés

La Fig. 2(a) montre les résultats des simulations sur la base Columbia modifiée avec des ensembles de chaînes sans boucle ni cycles pour différentes longueurs de chaînes avec le noyau de Kashima (Eq. 2). Lorsque des chaînes sont de longueur fixe ($|h| = k$), les performances sont d'autant meilleures que la longueur est petite, ce qui s'explique par le fait que les objets sont représentés par 1 à 3 régions. En utilisant des longueurs de chaînes variables ($|h| \leq k$), les performances ne sont pas améliorées bien que l'on considère davantage de possibilités. Cela peut s'expliquer par le fait que le noyau de Kashima noie les appariements intéressants dans la somme(cf section 2).

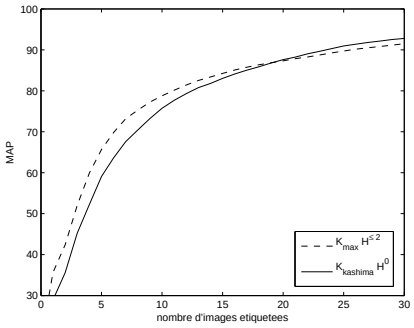
La Fig. 2(b) montre les résultats de simulations similaires aux précédentes, mais cette fois ci avec un noyau K_{max} (Eq. 5). On retrouve la même évolution avec des chaînes de longueur fixe ($|h| = k$), mais les différences sont plus grandes. Par contre, lorsque des chaînes de longueur variables sont utilisées ($|h| \leq k$), les résultats sont bien meilleurs. En effet la similarité entre les graphes avec ce noyau se résume à la similarité d'un seul couple de chaînes appariées. On augmente les chances de trouver ce meilleur appariement si on le recherche dans un ensemble de chaînes de plusieurs longueurs et non limité à une seule longueur.



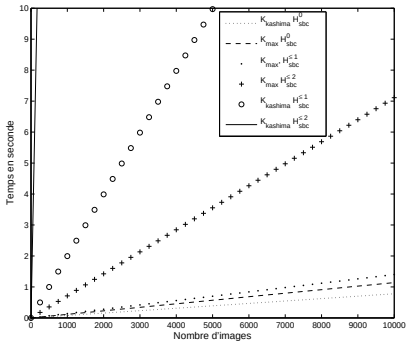
(a) Noyau $K_{Kashima}$



(b) Noyau K_{max}



(c) Comparaison des noyaux K_{max} et $K_{kashima}$ avec le meilleur paramétrage.



(d) Temps de calcul de la similarité d'une image avec toutes les autres de la base.

FIG. 2 – *Performance (%) sur la base Columbia modifiée avec le noyau $K_{Kashima}$ et K_{max}. L'ensemble des chaînes H_{sbc} sans boucle ni cycle est utilisé pour différentes longueurs de chaîne.*

4.2.2 Comparaison des deux noyaux

Si on compare le meilleur résultat obtenu avec le noyau K_{max} ($|h| \leq 2$) et le meilleur résultat avec le noyau $K_{Kashima}$ ($|h| = 0$) (Fig. 2(c)), on constate qu'au début de l'apprentissage (jusqu'à une vingtaine d'images annotées), le noyau K_{max} conduit à des résultats légèrement meilleurs que le noyau $K_{Kashima}$. Ensuite les deux courbes sont très proches. Par contre le temps de recherche du meilleur appariement avec l'algorithme de "branch and bound" est beaucoup plus rapide avec le noyau K_{max} qu'avec le noyau $K_{Kashima}$ (cf Fig. 2). Nous concluons que sur cette base où les objets d'intérêt sont représentés par au plus 3 régions, le noyau K_{max} calculé sur toutes les chaînes de longueur inférieure ou égale à 2 est préférable aux autres noyaux.

FIG. 3 – *Exemple de segmentation floue, les zones noires représentent les parties de l'image n'appartenant à aucune région. Il y a 16 régions dont 5 sur la tête.*

(a) (b)

FIG. 4 – *Recherche de visages sur la base Caltech à l'aide d'un noyau K_{max} et de l'ensemble de chaîne $H^{\leq 4}_{sbc}$. (a) Classement initial avec une annotation positive. (b) Classement avec 3 annotations positives et 2 négatives. La première erreur se trouve à partir de 70 images.*

4.3 Caltech

Nous utilisons la base Caltech avec la vérité terrain de la campagne d'évaluation PASCAL[4]. Cette base contient 5775 images réparties en 5 catégories de 450 à 1370 images. Les graphes issus de la segmentation en régions floues comprennent entre 1 et 23 sommets et le nombre moyen de sommets est d'environ 9 par image (cf. figure 3). Un exemple de session de recherche est donné sur la figure 4. Nous avons mené les expériences suivantes sur cette base :

- Attributs de régions floues, avec le noyau de Kashima (Eq. 2) et des ensembles de chaînes H^0. Nous n'avons pas considéré de chaînes plus longues pour des raisons de complexité.
- Attributs de régions floues, avec le noyau $K_{\max}$ (Eq. 5) et différents ensembles de chaînes.
- Attributs globaux sous la forme d'un histogramme de couleurs et de textures. Les histogrammes sont calculés sur un dictionnaire adapté à la base suivant un processus de quantification vectorielle décrit dans Gosselin (2005). Un noyau Gaussien avec une distance du χ^2 est utilisé.
- Attributs type points d'intérêt avec régions MSER (*cf.* Matas et al. (2002)) décrites par des SIFT (*cf.* Lowe (2004)). Dans ce cas, chaque image est représentée par un ensemble

[4] $http://www.pascal-network.org$

Catégorie	MSER	Global	$K_{kashima}\ H^0$	$K_{\max}\ H^{\leq 2}$	$K_{\max}\ H_e^{\leq 3}$	$K_{\max}\ H_{sbc}^{\leq 3}$
aeroplane	66	60	**92**	87	86	86
background	87	**90**	77	76	74	74
car	54	82	**97**	91	91	91
motorbike	59	78	**88**	68	68	68
face	70	77	**100**	84	84	84
Moyenne	67	77	**91**	81	81	81

TAB. 2 – *Précision Moyenne (%) sur chacune des catégories de la base Caltech, avec 50 annotations issues d'un processus d'apprentissage actif.*

de vecteurs SIFT correspondant à chaque zone d'intérêt. Un noyau de Kashima (Eq. 2) avec des sommets (H^0) est utilisé. Nous n'avons pas poussé les expériences sur des ensembles de chaînes plus importants étant donnée la complexité que cela engendrerait. En effet, il y a entre 50 et 150 vecteurs SIFT par image, ce qui rend les calculs déjà très longs en ne considérant que les sommets.

Les résultats de ces expériences sont présentées par catégories dans la table 2, et en fonction du nombre d'annotations dans la figure 5. Tout d'abord, si on s'intéresse au noyau Kashima et max avec le meilleur paramétrage sur la base Columbia modifiée, on peut remarquer que nous avons ici un comportement inverse. Sur la base Caltech, le noyau de Kashima avec H^0 est plus performant quelle que soit la catégorie. Cela peut s'expliquer par le fait que, sur cette base, le contexte joue un rôle important, contrairement à la base Columbia modifiée où il n'existe pas de relation particulière entre un objet et son fond. En effet, étant donné que ce noyau somme toutes les similarités entre les différents sommets, et que l'appariement entre les régions de fond, apporte une information pertinente. Par exemple, les voitures sont souvent en ville, dont les couleurs et les textures sont proches.

Puis, si on s'intéresse aux autres ensembles de chaînes pour le noyau max, nous pouvons remarquer qu'il n'y a pas de différence notable entre les différents ensembles. Cela peut s'expliquer par le fait que, sur la base Caltech, il existe un sous-ensemble de chaînes communes aux chaînes Eulérienne et sans boucle ni cycles qui sont suffisantes pour pouvoir discriminer les images. Ce résultat est intéressant en terme de complexité étant donné que, plus l'ensemble des chaînes considérées est petit, plus les calculs sont rapides.

Enfin, si l'on compare les résultats avec d'autres attributs que les régions floues, nous pouvons constater que les régions floues sont particulièrement intéressantes étant donné qu'elles permettent d'obtenir les meilleurs résultats, sauf la catégorie "background". Néanmoins, dans ce cas a priori défavorable (la catégorie "background" correspond à une recherche de type globale), les régions floues arrivent toutefois à donner de très bon résultats. Ceci tend à montrer la capacité des régions floues à pouvoir s'adapter plus facilement aux différents types de catégories, de la recherche globale à la recherche purement locale, en passant par les cas où le contexte peut jouer un rôle important.

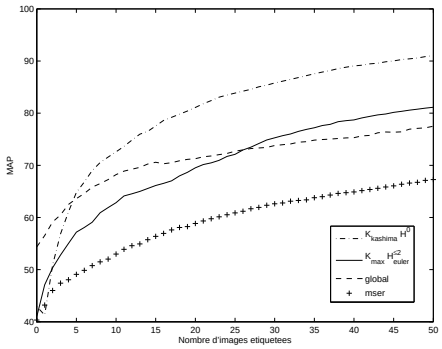


FIG. 5 – *Performances moyenne (%) sur les catégories de la base Caltech, pour différents attributs visuels et noyau sur graphes, en fonction du nombre d'annotations.*

5 Conclusion

Nous avons montré que l'utilisation de noyau de graphe dans le cadre de la recherche itérative d'objet était possible et efficace. D'après nos expériences, il semble que la description d'une image à l'aide de régions soit plus efficace qu'une description globale ou une description basée sur des points caractéristiques. En effet une primitive région, même si elle ne colle pas parfaitement à la sémantique de l'image porte une information locale plus robuste aux variations.

Les imprécisions de la segmentation et les différences d'aspect d'un objet d'une image à l'autre sont compensées par la mise en correspondance de graphes qui soit la plus générale possible. Les noyaux de graphes calculés à partir de noyaux sur des chaînes dans ces graphes permettent de trouver une solution optimale au problème de l'appariement des graphes avec un algorithme de branch and bound. Cependant, l'emploi des noyaux de Kashima, qui recherchent le couple de chaînes les plus semblables parmi tous les couples de chaînes possibles est incompatible avec l'utilisation en temps réel. Il faut réduire la longueur des chaînes à au plus deux sommets (pour des graphes qui comportent 10 à 20 régions). A longueur de chaînes égale, il vaut mieux prendre un noyau $K_{\max}$, qui, associé à l'algorithme de « branch and bound » trouve le meilleur appariement beaucoup plus rapidement que le noyau de Kashima. Sur les deux bases utilisées il s'est avéré que l'utilisation de petites chaînes suffisait à une bonne reconnaissance des objets. Et pour l'instant nous n'avons pas pu démontrer l'intérêt d'utiliser un ensemble particulier de chaînes.

Le choix de la longueur maximale des chaînes reste le problème principal, il y a un compromis à faire entre temps de calcul et efficacité. Il semble que cette longueur soit liée d'une part au nombre de régions formant l'objet et d'autre part à l'importance du fond dans la reconnaissance de l'objet. Dans le cas où le fond est important pour reconnaître un objet, un noyau basé sur les seules régions suffit. Par contre si l'objet doit être trouvé quel que soit le contexte, la structure du graphe est importante. Nous avons dans cette première étude, fait intervenir que l'adjacence entre les régions. L'emploi d'attributs plus précis de position relative, dans le noyau sur les arêtes, devraient améliorer considérablement la recherche, en la contraignant.

Références

Eichhorn, J. et O. Chapelle (2004). Object categorization with svm : kernels for local features. Technical report.

Gosselin, P. (2005). *Méthodes d'apprentissage pour la recherche de catégories dans des bases d'images*. Ph. D. thesis, Université de Cergy-Pontoise.

Gosselin, P. et M. Cord (2006). Precision-oriented active selection for interactive image retrieval. In *IEEE International Conference on Image Processing*, Atlanta, GA, USA.

Grauman, K. et T. Darell (2005). The pyramid match kernel : Discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision (ICCV)*, Beijing, China.

Jurie, F. et B. Triggs (2005). Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision and Pattern Recognition*.

Kashima, H. et Y. Tsuboi (2004). Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In *International Conference on Machine Learning (ICML)*, Banff, Alberta, Canada.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision (IJCV) 2*(60), 91–110.

Matas, J., O. Chum, M. Urban, et T. Pajdla (2002). Rosbut wide baseline stereo from maximally stable external regions. In *Proceedings of the Bristish Machine Vision Conference*, pp. 384–393.

Philipp-Foliguet, S. et J. Gony (2006). FReBIR : Fuzzy regions-based image retrieval. In *Information Processing and Management of Uncertainty (IPMU)*, Paris, France.

Sorlin, S., O. Sammoud, C. Solnon, et J.-M. Jolion (2006). Mesurer la similarité de graphes. In *Extraction de COnnaissance à partir d'Images (ECOI'06), Atelier de Extraction et Gestion de Connaissances (EGC'06)*, pp. 21–30.

Suard, F., V. Guigue, A. Rakotomamonjy, et A. Bensrhair (2005). Pedestrian detection using stereo-vision and graph kernels. In *Intelligent Vehicles Symposium*, Las Vegas, Nevada, pp. 267–272.

Wallraven, C., B. Caputo, et A. Graf (2003). Recognition with local features : the kernel recipe. In *International Conference on Computer Vision (ICCV)*, Volume 2, pp. 257–264.

Summary

In the framework of the interactive search in image databases, we are interested in similarity measures able to learn during the search and usable in real-time. Images are represented by adjacency graphs of fuzzy regions. In order to compare attributed graphs, we employ kernels on graphs built on sets of paths. We propose a general framework allowing the use of various kernels and various types of paths (without cycle, with loops) in order to perform inexact matchings. We achieved comparisons on two bases issued from Columbia and Caltech and showed that paths of very small dimension (length lower than 3) are the most effective to retrieve objet categories.