



HAL
open science

Recherche d'objets par appariement de graphes combinant contours et régions

Jean-Emmanuel Haugeard, Sylvie Philipp-Foliguet

► **To cite this version:**

Jean-Emmanuel Haugeard, Sylvie Philipp-Foliguet. Recherche d'objets par appariement de graphes combinant contours et régions. RFIA: Reconnaissance des Formes et Intelligence Artificielle, Jan 2010, CAEN, France. hal-00515880

HAL Id: hal-00515880

<https://hal.science/hal-00515880>

Submitted on 8 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Recherche d'objets par appariement de graphes combinant contours et régions

J.-E. HAUGEARD¹

S. PHILIPP-FOLIGUET¹

¹ ETIS, CNRS, ENSEA, Univ. Cergy-Pontoise

6 avenue du Ponceau, BP 44, F95014 Cergy Pontoise, France
{jean-emmanuel.haugeard, sylvie.philipp}@ensea.fr

Résumé

Cette dernière décennie, la modélisation des villes 3D est devenue l'un des enjeux de la recherche multimédia et un axe important en reconnaissance d'objets. Nous nous sommes intéressés à localiser différentes primitives, plus particulièrement les fenêtres, dans les façades de Paris. Une image de contours est lisible par l'oeil humain qui effectue un groupement perceptuel et distingue les entités présentes dans la scène. C'est ce mécanisme que nous avons cherché à reproduire. L'image est représentée sous la forme d'un graphe d'adjacence de segments de contours, valué par des informations d'orientation et de proximité des segments de contours. Pour la mise en correspondance inexacte des graphes, nous proposons plusieurs variantes d'une nouvelle similarité basée sur des ensembles de chemins tracés sur les graphes, capables d'effectuer les groupements des contours et robustes aux changements d'échelle. La similarité entre chemins prend en compte la similarité des ensembles de segments de contours et la similarité des régions définies par ces chemins. La sélection des images d'une base contenant un objet particulier s'effectue à l'aide d'un classifieur SVM ou kppv. La localisation des objets dans l'image utilise un système de vote à partir des chemins sélectionnés par l'algorithme d'appariement. Nos résultats montrent que l'on est capable d'extraire les fenêtres des façades à partir d'un ou plusieurs exemples.

Mots Clef

Graphe Relationnel Attribué, appariement inexact de graphes, similarité de graphes, noyau sur graphe, fenêtre, façade

Abstract

In the last decade, the 3D modeling of cities has become one of the challenges of multimedia search and an important focus in object recognition. We are interested in locating various primitives, especially windows, in Paris façades. An image of edges is readable by the human visual system, which performs a perceptual grouping and is thus

able to catch the entities of the image. We tried in this paper to model this mechanism. Images are represented as adjacency graphs of contour segments, attributed with orientation and proximity information. To achieve the graph matching, we propose several variants of a new similarity based on sets of paths carried by the graphs, able to perform the perceptual grouping and robust to scale changes. The similarity between paths takes into account the similarity between sets of edges and the similarity between regions defined by these paths. The retrieval of images containing a specific object is performed thanks to a SVM or knn classifier. For object localization, we use a vote system from the paths selected by the matching algorithm. Our results show that we are able to extract the windows from images of façades from examples.

Keywords

Attributed Relational Graph, inexact graph matching, graph similarity, graph kernel, window, facade

1 Introduction

Grâce à l'apparition ces dernières années de différents navigateurs 3D géographiques (Google Street View, Geoportail, Microsoft live Earth), la modélisation des villes connaît un regain d'intérêt. Plusieurs approches de modélisation de l'environnement urbain ont été axées sur une modélisation grossière, par exemple : une représentation polyédrique des murs, du toit, du sol... Néanmoins, les derniers axes de recherche tentent d'analyser les images réelles des façades en extrayant les fenêtres, les portes et les ornements afin de fournir des informations riches sur les bâtiments et ainsi ajouter du réalisme à la visualisation [12, 1, 10]. Dans ce contexte, la détection et la classification de primitives (fenêtres, portes, enseignes, voitures...) peut être bénéfique pour améliorer la navigation dans ces villes et les différents services mis en place.

L'un des objectifs du projet iTOWNS¹, dans lequel s'inscrit ce travail, est de mettre en place un moteur de recherche

¹Projet ANR (ANR 07-MDCO-007-03)

basé sur une indexation des données images de manière à proposer au sein même du navigateur des services basés sur des requêtes (trouver la localisation d'une image apportée par l'utilisateur, sélectionner les images contenant telle enseigne, rechercher des immeubles de moins de n étages de type Haussmannien, etc.). Afin de caractériser les différentes images de façades, nous nous sommes donc concentrés sur la structure des façades et l'extraction d'entités architecturales.

Dans cet article, nos travaux ont essentiellement porté sur l'extraction des fenêtres dans les façades. Les fenêtres constituent en effet la primitive à la fois constituante et structurante des façades. De plus cette primitive est relativement simple et nous permet de développer des algorithmes que nous pourrions ensuite étendre à d'autres primitives telles que les voitures, piétons etc. En nous basant sur le groupement perceptuel des contours [4], nous proposons de représenter les façades par des graphes relationnels attribués de segments de contours et de définir une similarité basée sur les graphes de contours.

Lors de travaux antérieurs [5], nous avons proposé des similarités entre graphes basées sur des similarités entre chemins définis sur ces graphes. Ces similarités sont rappelées en section 2. Nous proposons en section 3 de nouvelles similarités prenant en compte non seulement les contours, mais aussi l'information radiométrique des régions délimitées par les chemins. L'appariement des graphes s'effectue par l'algorithme de "branch and bound" qui permet la mise en correspondance inexacte des graphes (y compris de graphes d'ordres différents). De plus, le calcul incrémental de nos fonctions de similarité accélère le temps de calcul et autorise des calculs approchés.

Ces nouvelles similarités permettent de nouvelles applications plus générales. Nous nous affranchissons dorénavant de l'étape de sélection d'hypothèses de fenêtres comme dans [6, 9]. Les fenêtres sont extraites directement des façades par recherche des sous-graphes les plus similaires à des graphes exemples de fenêtres. Un système de vote basé sur la similarité permet une localisation précise des objets recherchés dans les images.

2 Appariement de graphes de contours

La notion de contours semble être un axe pertinent pour représenter l'information architecturale des façades. De récents travaux proposent plusieurs approches pour construire des descripteurs basés contours dédiés à un objet spécifique. Shotton et al. [15] et Opelt et al. [13] apprennent une distribution de contours de l'objet. Plus récemment, Ferrari et al. [3, 4] ont utilisé les propriétés de groupement perceptuel des contours en proposant un algorithme basé sur les parcours possibles dans une représentation réseau de l'image. Ils proposent une nouvelle méthode basée sur des paires de segments adjacents pour apprendre un modèle de forme d'un objet. Nous nous sommes donc orientés sur cette notion de groupement perceptuel de

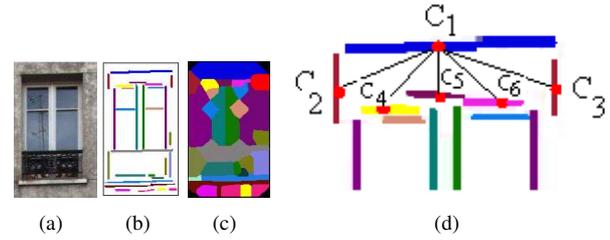


FIG. 1 – **Extraction et graphe de segments de contours** (a) Exemple d'une fenêtre. (b) Segments de contours extraits et polygonalisés. (c) Diagramme de Voronoi sur les segments de contours. (d) Zoom sur une partie du graphe. C_1 est relié par une arête à C_2, C_3, C_4, C_5, C_6 , mais pas aux autres segments non adjacents dans le diagramme de Voronoi.

contours et avons représenté les images par un Graphe Relationnel Attribué.

2.1 Représentation de l'objet en Graphe Relationnel Attribué

Pour chaque image, les segments de contours sont détectés à l'aide des opérateurs de Shen-Castan [14], puis prolongés et polygonalisés. Afin de considérer l'ensemble des segments comme une structure, nous représentons cet ensemble par un graphe relationnel attribué (ARG). Chaque segment de contour C_i est représenté par un noeud v_i du graphe et l'arête e_{ij} du graphe représente la position relative entre deux segments de contours C_i, C_j . L'information topologique (parallélisme, proximité) peut être considérée uniquement pour les voisins proches des segments de contours. Afin de trouver les segments de contours les plus proches, nous utilisons le diagramme de Voronoi. Une arête dans le graphe relationnel représente la relation entre deux segments proches dans le sens où leurs cellules de Voronoi sont adjacentes.

Afin d'être robuste aux changements d'échelle, nous avons choisi de caractériser un segment de contour uniquement par son orientation (horizontal, verticale...). Si Θ désigne l'angle entre le segment de contour C_i et la ligne horizontale ($\Theta \in [0, 180[$), C_i est alors représenté dans le graphe par le noeud $v_i = (\cos(2\Theta), \sin(2\Theta))^T$.

L'arête du graphe $e_{ij} = (v_i, v_j)$ représente la proximité entre le segment de contour C_i and C_j (si C_i and C_j ne sont pas proches au sens de Voronoi, e_{ij} n'existe pas). L'arête dans le graphe est caractérisée par la position relative des centres de gravité des segments de contours C_i and C_j , notés $g_{C_i}(Xg_{C_i}, Yg_{C_i})$ et $g_{C_j}(Xg_{C_j}, Yg_{C_j})$. L'arête est donc affectée d'un vecteur $e_{ij} = (Xg_{C_j} - Xg_{C_i}, Yg_{C_j} - Yg_{C_i})^T$.

2.2 Appariement inexact de graphes de contours

Le problème de l'extraction d'entités d'architecturales peut-être considéré comme un problème d'appariement inexact de graphes à partir d'un graphe requête. La problématique de la recherche d'appariement inexact de graphes valués est double, définir une mesure de similarité entre graphes de tailles différentes, et trouver les meilleurs appariements en un temps raisonnable. Dans cet article, nous nous sommes intéressés à la partie mesure de similarité plutôt qu'à la complexité en temps. Nous avons cependant utilisé des approximations autorisées par l'algorithme de "branch and bound" pour réduire cette complexité. Nous verrons que cet algorithme est compatible avec nos mesures de similarités et permet les approximations.

Pour la recherche d'appariement de graphes, des approches récentes de comparaison de graphes [8, 7] proposent de considérer le graphe comme un ensemble de chemins.

Un chemin h dans un graphe $G = (V, E)$ est une suite de noeuds de V reliés par des arêtes de E : $h = (v_0, v_1, \dots, v_n)$, $v_i \in V$. On notera $|V|$ le cardinal de V et $|h|$ la longueur du chemin h , ie son nombre d'arêtes.

Kashima et al. ont proposé [7] des similarités employant des noyaux sur des graphes pour comparer les graphes G et G' entre eux. Leurs noyaux se ramènent à des noyaux sur chemins, en effet ils proposent de comparer tous les chemins possibles de même longueur entre les deux graphes. Le problème de ce noyau est la forte complexité en temps, acceptable pour les petits graphes de molécules biochimiques, mais inabordable avec nos grands graphes attribués. D'autres noyaux sur graphes furent proposés par Lebrun et al. [8], qui sont plus rapides que le noyau de Kashima.

En se basant sur les noyaux sur graphes de régions de [8], nous avons conçu un noyau sur graphes de contours. Ce noyau prend en compte tous les sommets des graphes, en considérant un chemin commençant à chaque noeud :

$$K_{struct}(G, G') = \frac{1}{|V|} \sum_{i=1}^{|V|} \max_{h_{v_i}, h'} K_C(h_{v_i}, h') + \frac{1}{|V'|} \sum_{i=1}^{|V'|} \max_{h, h'_{v'_i}} K_C(h, h'_{v'_i}) \quad (1)$$

avec h_{v_i} un chemin débutant par le noeud v_i . Notons que cette fonction ne respecte pas les conditions de Mercer (matrice définie positive), 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. K_C représente un noyau sur chemin. Divers K_C furent présentés [8] (somme, produit...). Nous avons testé ces divers noyaux et les meilleurs résultats furent obtenus avec le K_C ci-dessous, où e_j représente (v_{j-1}, v_j) :

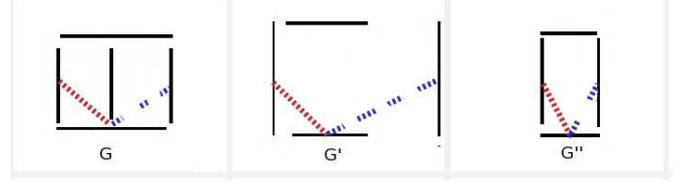


FIG. 2 – Exemples de graphes avec un chemin de taille 2 en pointillé. La pénalité d'échelle diminuera la similarité de G' avec G

$$K_C(h_{v_i}, h') = K_v(v_i, v'_0) + \sum_{j=1}^{|h|} K_e(e_j, e'_j) K_v(v_j, v'_j)$$

K_v et K_e sont les noyaux mineurs qui définissent la similarité entre noeuds (orientation des segments de contours) et la similarité entre arêtes (positions relatives entre deux contours proches). Nous avons proposé ces deux noyaux :

$$K_e(e_j, e'_j) = \frac{\langle e_j, e'_j \rangle}{\|e_j\| \cdot \|e'_j\|} + 1$$

et

$$K_v(v_j, v'_j) = \frac{\langle v_j, v'_j \rangle}{\|v_j\| \cdot \|v'_j\|} + 1$$

$\|\cdot\|$ étant la norme euclidienne.

L'objectif de notre noyau est de comparer un ensemble de contours, du point de vue de leur orientation et de leur position relative. Certains chemins n'apportent aucune information structurelle, car formés par des segments de contours parallèles (ex : corniche, toit... cf. figure 4 a avec son balcon). Afin de traiter ce problème, nous pouvons augmenter la taille de chemins, mais la complexité en temps devient vite inabordable. C'est pourquoi nous avons proposé d'ajouter un poids $O_{i,j}$ dans le noyau K_C qui pénalise les chemins dont l'orientation des segments de contours ne varie pas.

$$O_{i,j} = \sin(\phi_{ij}) = \sqrt{\frac{1}{2}(1 - \langle v_i, v_j \rangle)}$$

avec ϕ_{ij} l'angle entre le contour C_i et C_j .

De plus le groupement perceptuel d'un ensemble de contours est un point important dans la reconnaissance d'objet. Par exemple, sur la figure 2, nous pouvons nous poser la question de savoir si l'on considère le contour le plus à droite du graphe G' comme un contour de l'objet ou non. Afin de modéliser cette information, nous avons ajouté une pénalité S_{e_i} .

$$S_{e_i} = \min\left(\frac{\|e_i\|}{\|e_{i-1}\|} \cdot \frac{\|e'_{i-1}\|}{\|e'_i\|}, \frac{\|e_{i-1}\|}{\|e_i\|} \cdot \frac{\|e'_i\|}{\|e'_{i-1}\|}\right)$$

Bien que $O_{i,j}$ et S_{e_i} soient illustrés dans le cas des graphes de fenêtres, ces poids restent pertinents pour d'autres objets car aucune information perceptuelle ne peut être extraite d'un ensemble de contours tous parallèles.

Notre noyau final K_C devient ($Se_i \in [0, 1]$ et $O_{i,j} \in [0, 1]$) :

$$K_C(h_{v_i}, h'_{v'_j}) = K_v(v_i, v'_j) + \sum_{j=1}^{|h|} Se_j O_{j,j-1} K_e(e_j, e'_j) K_v(v_j, v'_j) \quad (2)$$

2.3 Application à la classification de graphes

Pour évaluer ce noyau et ses différentes pondérations, nous avons fabriqué une base de 300 imagerie extraites d'images de façades de Paris. Ces imagerie ont été extraites à l'aide d'un algorithme basé sur les histogrammes des gradients verticaux et horizontaux ([6, 9]) et constituent des hypothèses de fenêtres. Après extraction des graphes de contours, nous entraînons un classifieur SVM (séparateurs à vaste marge) dans le but de classer ces imagerie soit en fenêtre soit et négatifs urbains : gouttière, corniche...), avec un ordre de pertinence.

Nous évaluons notre noyau, l'intérêt des pondérations proposées et les comparons avec une méthode de référence, basée sur des points d'intérêt (détecteurs de région MSER et les descripteurs SIFT [11]). Les noyaux nous permettent aussi d'utiliser des techniques d'apprentissage actif qui sélectionnent les meilleures images à faire annoter par l'utilisateur. Chaque session simule une recherche où une requête est choisie au hasard parmi les imagerie de fenêtres et annotée positivement. Un premier classement est obtenu, exclusivement basé sur la similarité. Puis, les images sélectionnées par la technique d'apprentissage actif sont annotées en fonction de la catégorie choisie (ici les fenêtres). La simulation annoté (positivement et négativement) quelques images à chaque itération. Le système met à jour le classement de la base de données en fonction de ces nouvelles étiquettes. L'ensemble du processus est itéré 100 fois avec des images initiales différentes et la Précision est moyennée sur toutes ces sessions pour donner le MAP (Mean Average Precision) (figure 3). Le graphique met clairement en évidence l'apport des deux pondérations. Notons qu'avec un seul exemple de fenêtre et un seul contre-exemple, 80 % de la base est bien classée avec notre noyau pondéré, contre 70 % avec la seule pondération pénalisant le parallélisme des contours et 50 % avec les points d'intérêt. Avec 10 exemples positifs et 10 exemples négatifs, on obtient 85 % de classement correct. Suite à plusieurs test sur la longueur de h , les meilleurs résultats sont obtenus avec $|h| = 8$.

Si à cause de la présence du "*max*" dans le noyau que nous proposons (équation 1) celui-ci ne vérifie pas de manière théorique les conditions de Mercer, ce "*max*" est intéressant quand il est recherché par l'algorithme de "branch and bound". En effet dans ce cas, l'ensemble des solutions possibles de problème de l'appariement est représenté par un arbre de recherche. Chaque noeud de cet arbre représente un couple de sommets $(v, v') \in V \times V'$ candidats à l'appariement. L'algorithme "branch and bound" permet

de trouver la 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 $K(G, G')$. Puis les autres branches de l'arbre sont explorées 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. On peut réduire le temps de calcul si on se contente d'une solution sous-optimale.

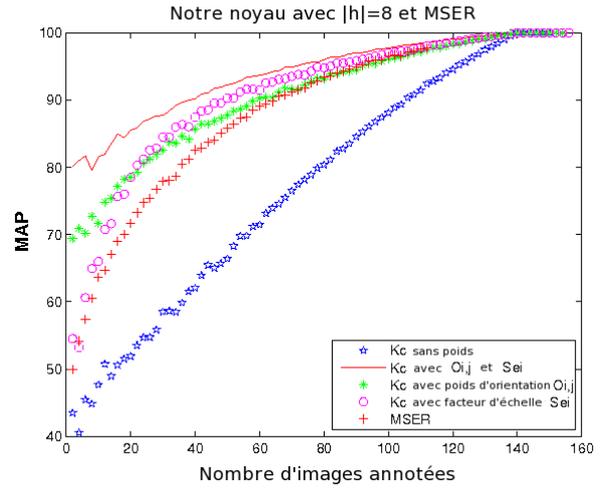


FIG. 3 – Évaluation de notre noyau K_{struct} et de ses pondérations lors de la classification d'imagerie de façades.

3 Appariements mixant contours et régions

Dans la section précédente, nous avons introduit et utilisé les graphes globaux de contours pour classer des imagerie contenant des hypothèses de fenêtres selon leurs similarités. Dans cette section nous voulons nous affranchir de l'étape d'extraction d'hypothèses de fenêtres et exploiter directement le graphe des façades pour localiser la position des sous-graphes fenêtres. Nous cherchons donc à appairer les graphes requêtes avec des sous-graphes des façades. Ceci nous permettra non seulement de classer les façades en fonction de la fenêtre requête mais aussi de localiser précisément les fenêtres dans les façades. Le premier aspect répond à une requête du type "trouver des façades de type Haussmannien", à partir d'un exemple ou d'un ensemble d'exemples de fenêtres Haussmanniennes. La localisation des fenêtres a des applications pour la modélisation ou la reconstruction procédurale des bâtiments.

Nous utiliserons la notion de "chemin d'intérêt", qui est un chemin du graphe cible G' apparié avec un chemin du graphe requête G . Un chemin d'intérêt déterminera une "région de chemin d'intérêt" que nous définirons dans le paragraphe suivant. Nous pourrions alors introduire des similarités exploitant à la fois la similarité des contours et

des régions d'intérêt définies par ces contours.

3.1 Région de chemin d'intérêt

Dans la similarité définie par l'équation(1), pour chaque noeud v_i de G , nous cherchons le meilleur appariement entre le chemin $h_{v_i} \in G$ (chemin débutant par le noeud v_i) et un chemin $h' \in G'$ (cf. figure 4).

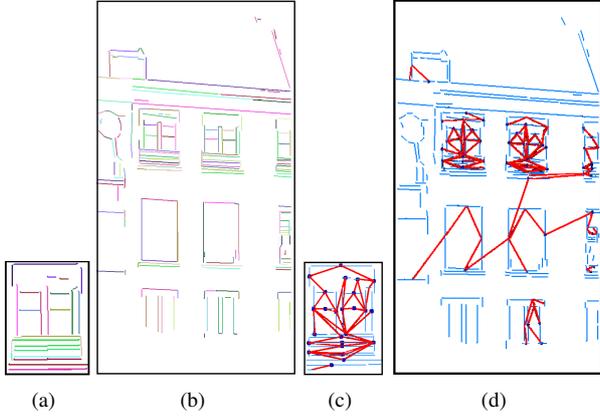


FIG. 4 – **Les meilleurs appariements** (a) Graphe requête. (b) Graphe façade (pour une meilleure lisibilité, les arêtes ne sont pas tracées). (c)(d)(traits rouges) Chemins appariés du sous-graphe requête (a) et du graphe (b). On constate que la majorité des chemins est localisée sur des régions pertinentes.

A partir du meilleur h' retrouvé, nous définissons une région de chemin d'intérêt (RCI) autour de ce chemin h' . Nous définissons RCI, le rectangle dont les côtés sont parallèles aux bords de l'image et qui contient tous les centres de gravité des segments de contours appartenant au chemin. Afin de localiser les fenêtres dans le graphe des façades, nous affectons une valeur d'évaluation à chaque pixel de la RCI. Cette valeur peut être un simple vote (+1 à chaque pixel de la RCI) ou un vote pondéré (par la similarité de l'appariement). Tous les meilleurs chemins h' appariés à un chemin h_{v_i} de G vont donc voter pour un ensemble de pixels dans G' . Le résultat (figure 5) est une accumulation de votes de ces régions de chemins d'intérêt. Pour chaque pixel (x,y) de l'image, nous obtenons :

$$J_{G,G'}(x,y) = \sum_{i=1}^{|V|} J_{h_{v_i},h'}(x,y) \quad (3)$$

où $J_{h_{v_i},h'}(x,y)$ vaut la valeur d'évaluation. Dans notre cas, nous avons choisi un vote pondéré $J_{h_{v_i},h'}(x,y) = K_C(h_{v_i},h')$ où $K_C(h_{v_i},h')$ est la valeur de similarité entre le chemin h_{v_i} et h' .

De plus, nous souhaitons mesurer la similarité d'un graphe de fenêtre comme sous-graphe dans les graphes de façades. Or dans l'équation 1, seul le premier terme nous intéresse :

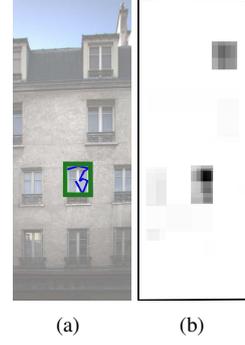


FIG. 5 – **Localisation des fenêtres** (a) Un chemin et sa région d'intérêt. (b) Image J obtenue par accumulation des régions d'intérêt.

$$\frac{1}{|V|} \sum_{i=1}^{|V|} \max_{h_{v_i},h'} K_C(h_{v_i},h').$$

Ainsi, nous avons décidé de ne plus respecter la symétrie et de seulement mesurer la similarité dans un sens (trouver un graphe de fenêtre dans un graphe de façade). N'étant plus symétrique, nous avons pris comme classifieur un k-Plus Proche Voisin.

Dans [5], nous avons mis en évidence le besoin d'avoir de nombreux chemins pour décrire l'objet et atténuer les chemins de bruit. Pour cela, on choisissait plusieurs chemins de taille fixe. Notons que selon l'application, la longueur des chemins peut être très variable. Kashima [7] sommait sur toutes les longueurs de chemins tracés sur des molécules chimiques. Lebrun [8] a montré que des chemins de longueur 1 étaient suffisants pour des graphes de régions. Dans le premier cas, l'information portée par les noeuds est très réduite (un symbole chimique), dans le deuxième cas, elle est très riche (couleur et texture). Dans notre cas, nous proposons maintenant de rechercher les meilleurs chemins débutant par chaque noeud mais dont la taille de chemin varie. L'algorithme de "branch and bound" permet d'explorer les branches de l'arbre de recherche les plus intéressantes en terme de similarité. Ainsi si l'arbre est exploré jusqu'à une profondeur l , les chemins intéressants aux profondeurs inférieures ont déjà été explorés par le "branch and bound". Nous ne sommes donc plus obligés de réexplorer l'arbre pour les profondeurs inférieures. Par récursivité, on peut rapidement retrouver ces chemins. Notre similarité devient :

$$K_{struct}(G,G') = \frac{1}{n_a} \sum_{i=1}^{|V|} \sum_{|h|=3}^{|h|=l} \max_{h_{v_i},h'} K_C(h_{v_i},h') \quad (4)$$

avec

- n_a nombre d'appariements de chemins retournés
- l longueur maximale des chemins.

On cherche ainsi pour un noeud v_i , le meilleur chemin commençant par celui-ci pour une taille 3 puis 4 ... jusqu'au chemin de taille l . On considère ainsi que les chemins de bruit sont uniformément répartis. Ainsi pour les meilleurs appariements, on a de la redondance. En effet, par exemple, si le chemin de taille l est un bon appariement alors la plupart du temps, on constate que le chemin de taille $l+1$ est une extension du chemin de taille l . Au contraire, si le chemin de taille l trouvé est un faux positif, alors le chemin de taille $l+1$ est souvent différent du chemin précédent.

Nous évaluons notre similarité (équation 4) sur une base composée de 200 images prises dans Paris² dont 60 façades. Nous choisissons de faire varier la longueur de h de 3 à 8 ($3 \leq |h| \leq 8$). Pour un graphe requête fenêtre (imagerie en haut à gauche), le système classe les images par ordre de similarité. Les meilleures images retournées (figure 6) avec la similarité (équation 4) sont des façades contenant ce style de fenêtre.



FIG. 6 – Une fenêtre requête et les images retournées par ordre de similarité sur l'appariement de chemins de contours. Les images sont classées de gauche à droite, puis de haut en bas. La façade au rang 1 contient la requête. Les façades signalées en rouge (rangs 5,6,13) sont des faux positifs. Les autres façades contiennent des fenêtres du même style que la requête.

²Les images furent acquises à l'aide du système mobile STEREOPOPLIS de l'IGN. Copyright images : ©IGN pour le projet iTOWNS.

3.2 Approche mélangant information contours et régions

Malgré l'adjonction du coefficient S_{e_i} (équation 2), le groupement des contours ne s'effectue pas toujours bien. Pour pallier ce problème, nous introduisons de l'information région (texture/couleur...) combinée à l'information portée par les contours. Maintenant que nous obtenons des régions d'intérêt autour des meilleurs chemins, nous pouvons évaluer ces régions en utilisant des descripteurs régions. Nous proposons deux approches de combinaisons entre l'information régions et contours. La première approche consiste à sélectionner les meilleurs appariements de chemins uniquement sur l'information contours, puis d'ajouter l'information région sur ces meilleurs appariements. La deuxième approche consiste à sélectionner les meilleurs appariements de chemins à l'aide de l'information contours et régions. Une nouvelle similarité de chemins mixant les deux informations est proposée.

Approche contours, puis régions.

Lors du calcul de similarité entre graphes (équation 4), nous cherchons, pour chaque chemin h_{v_i} du graphe requête, le chemin h' le plus similaire. Le chemin h' définit une région RCI, que l'on peut comparer à la RCI définie par h_{v_i} . Nous proposons la similarité suivante qui mixe les similarités des chemins (basées contours) et des régions :

$$K_{struct}(G, G') =$$

$$\frac{1}{n_a} \sum_{i=1}^{|V|} \sum_{|h|=3}^{|h|=l} S_R(B_{h_{v_i}}, B_{h'}) \max_{h_{v_i}, h'} K_C(h_{v_i}, h')$$

et l'accumulation de votes devient :

$$J_{h_{v_i}, h'}(x, y) = S_R(B_{h_{v_i}}, B_{h'}) K_C(h_{v_i}, h_{v_i})$$

avec :

- $B_{h_{v_i}}$ (respectivement $B_{h'}$) la région d'intérêt englobant le chemin h_{v_i} (resp. h').
- $S_R(B_{h_{v_i}}, B_{h'})$ la similarité entre les deux régions d'intérêt $B_{h_{v_i}}$ et $B_{h'}$.
- $K_C(h_{v_i}, h')$ similarité sur graphes de contours (équation 2).

Lors de nos tests (cf figure 7), nous avons choisi une similarité texture entre les régions. Nous utilisons les histogrammes basés sur les ondelettes quaternioniques [2] et calculons la similarité à l'aide d'un noyau gaussien et une distance du χ^2 .

$$S_R(B_{h_{v_i}}, B_{h'}) = \exp^{-\frac{1}{2} \chi^2(H, H')}$$

$$\chi^2(H, H') = \sum_i \frac{(H[i] - H'[i])^2}{H[i] + H'[i]}$$

avec :

H et H' les histogrammes sur les régions $B_{h_{v_i}}$ et $B_{h'}$.



FIG. 7 – Une requête et les images retournées par ordre de similarité basée sur les contours, puis sur les régions. La façade au rang 1 contient la requête. Les façades signalées en rouge (rangs 15,17) sont des faux positifs. Les autres façades contiennent des fenêtres du même style que la requête.

Approche contours et régions.

Dans la première approche, on cherchait les meilleurs appariements de chemins basés sur une similarité contours, puis on affinait les résultats sur ces meilleurs appariements avec une similarité entre régions. Dans cette deuxième approche (cf figure 8), nous proposons de rechercher les meilleurs appariements de chemins basés sur une similarité mixant contours et régions. Le choix du chemin ne se fait plus uniquement sur la suite de contours mais aussi sur la région définie par ces contours. Nous proposons de garder l'équation 4 mais notre K_C devient :

$$K_C(h_{v_i}, h') = K_v(v_i, v'_0) + \sum_{j=1}^{|h|} S_R(B_{h_{v_i}} B_{h'_j}) K_e(e_j, e'_j) K_v(v_j, v'_j)$$

3.3 Apprentissage à l'aide d'un classifieur $kppv$

En ce qui concerne la localisation des fenêtres, seules 2 ou 3 sont détectées sur la figure 5. Cela est dû au fait que chaque noeud du graphe requête ne vote que pour un



FIG. 8 – Une requête et les images retournées par ordre de similarité conjointe sur les contours et les régions. La façade au rang 1 contient la requête. Les autres façades contiennent des fenêtres du même style que la requête. Il n'y a plus de faux positifs dans le top 20.

seul chemin du graphe cible. Les chemins appariés dans le graphe cible peuvent se "répartir" sur plusieurs fenêtres (cf figure 4), réduisant la détection de celles-ci mais aussi oblitérant la détection des autres. Afin de généraliser et de trouver plus de fenêtres, nous proposons d'apprendre un ensemble \mathbb{A} de fenêtres du même style. Pour cela, pour chaque image de la base, nous calculons la similarité avec chacune des images requêtes et ne conservons que les k plus grandes. La valeur de similarité par rapport à l'ensemble d'apprentissage \mathbb{A} est :

$$K(G') = \frac{1}{k} \sum_{i=1}^k K_{struct}(G_i, G')$$

où

- $K_{struct}(G_i, G')$ est la similarité définie par équation (4).
- $G_i \in \mathbb{A}_k$ graphe annoté.
- \mathbb{A}_k ensemble des k plus proches voisins entre le graphe G et les graphes annotés avec la similarité (équation 4).

et l'accumulation devient :

$$J'_G(x, y) = \frac{1}{k} \sum_{i=1}^k y_i J_{G_i, G'}(x, y)$$

où

$J_{G_i, G'}$ accumulation sur le graphe G' par rapport à la requête G (équation 3).

Les figures 9 et 10 montrent que la combinaison des requêtes permet de localiser plus de fenêtres.

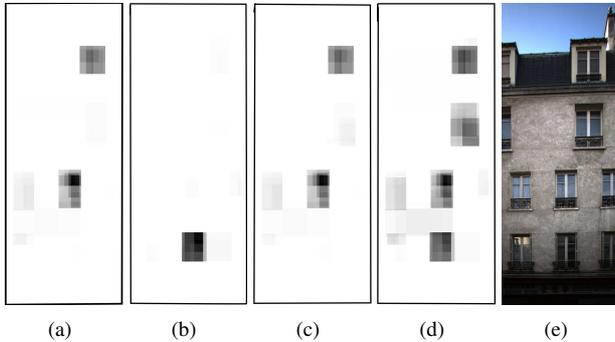


FIG. 9 – Accumulation sur une ensemble d'apprentissage de 3 fenêtres à 2 montants et un balcon (a) (b) (c) Accumulation obtenue avec 3 requêtes de même style. (d) Combinaison des 3 requêtes.

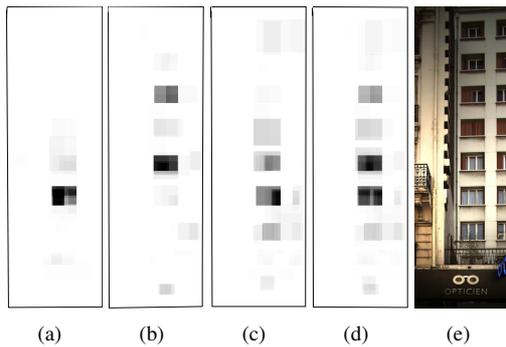


FIG. 10 – Accumulation sur une ensemble d'apprentissage de 3 fenêtres à 3 montants (a) (b) (c) Accumulation obtenue avec 3 requêtes de même style. (d) Combinaison des 3 requêtes.

4 Conclusion

Nous avons montré dans cet article l'intérêt d'exploiter les contours pour effectuer la recherche d'objets dans une base d'images mais aussi pour localiser ces objets dans l'image. La principale difficulté est d'obtenir le regroupement des contours appartenant à l'objet recherché. Pour cela, nous avons proposé de nouveaux noyaux, par ailleurs robustes aux changements d'échelle. Cependant, il a été nécessaire d'ajouter une information issue de la radiométrie des images pour renforcer la similarité des ensembles de contours. Des similarités mixant informations contours et régions délimitées par ces contours ont été proposées. L'algorithme de "branch and bound" employé avec ces nouveaux noyaux permet de trouver des solutions exactes

ou approchées au problème de l'appariement de graphes avec des graphes ou des sous-graphes d'autres images. Intégrées dans des classificateurs de type SVM ou kppv, les similarités proposées permettent de retrouver (et de classer) des images comme contenant un objet particulier décrit par l'exemple. Le noyau sur chemins proposé permet en outre une localisation de l'objet ou des différentes occurrences de l'objet dans l'image. Les méthodes et similarités proposées dans cet article étant très générales, elles seront exploitées sur d'autres types d'objets.

Remerciements.

Ce travail est financé par l'ANR dans le cadre du projet de recherche iTOWNS (ANR 07-MDCO-007-03).

Références

- [1] H. Ali, C. Seifert, N. Jindal, L. Paletta, and G. Paar. Window detection in facades. In *ICIAP*, 2007.
- [2] W.L. Chan, H. Choi, and R. Baraniuk. Quaternion wavelets for image analysis and processing. In *ICIP*, volume 5, pages 3057–3060.
- [3] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1):36–51, 2008.
- [4] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *IJCV à paraître*, 2009.
- [5] J.-E. Haugeard, S. Philipp-Foliguet, and F. Precioso. Windows and facades retrieval using similarity on graph of contours. In *ICIP*, November 2009.
- [6] J.-E. Haugeard, S. Philipp-Foliguet, F. Precioso, and J. Lebrun. Extraction of windows in facade using kernel on graph of contours. In *16th SCIA*, June 2009.
- [7] H. Kashima and Y. Tsuboi. Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In *ICML*, 2004.
- [8] J. Lebrun, S. Philipp-Foliguet, and P.-H. Gosselin. Image retrieval with graph kernel on regions. In *ICPR*, dec 2008.
- [9] S.C. Lee and Ramakant Nevatia. Extraction and integration of window in a 3d building model from ground view image. In *CVPR*, pages 113–120, 2004.
- [10] H. Mayer and S. Reznik. Building facade interpretation from uncalibrated wide-baseline image sequences. *PandRS*, 61(6):371–380, 2007.
- [11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65:43–72, 2005.
- [12] P. Müller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. In *ACM SIGGRAPH*, volume 26, 2007.
- [13] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages 575–588, 2006.
- [14] Castan S. Shen, J. An optimal linear operator for step edge detection. *GMIP*, 54(1):112–133, 1992.
- [15] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages 503–510, 2005.