



HAL
open science

Windows and Facades Retrieval using Similarity on Graph of Contours

Jean-Emmanuel Haugeard, Sylvie Philipp-Foliguet, Frédéric Precioso

► **To cite this version:**

Jean-Emmanuel Haugeard, Sylvie Philipp-Foliguet, Frédéric Precioso. Windows and Facades Retrieval using Similarity on Graph of Contours. IEEE International Conference on Image Processing (ICIP 09), Nov 2009, Cairo, Egypt. hal-00536261

HAL Id: hal-00536261

<https://hal.science/hal-00536261v1>

Submitted on 15 Nov 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.

WINDOWS AND FACADES RETRIEVAL USING SIMILARITY ON GRAPH OF CONTOURS

Jean-Emmanuel Haugeard, Sylvie Philipp-Foliguet, Frédéric Precioso

ETIS, CNRS, ENSEA, Univ Cergy-Pontoise,
6 avenue du Ponceau, BP 44,F 95014 Cergy Pontoise, France
{jean-emmanuel.haugeard, sylvie.philipp, frederic.precioso}@ensea.fr

ABSTRACT

The development of street-level geoviewers become recently a very active and challenging research topic. In this context, the detection, representation and classification of windows can be beneficial for the identification of the respective facade. In this paper, a novel method for windows and facade retrieval is presented. This method, based on a similarity of graph of contours, introduces a new kernel on graph for inexact graph matching. We design a kernel similarity function for structured sets of contours which will take into account the variations of contour orientation inside a structure set, as well as spatial proximity. Then we are able to extract a window as a sub-graph of the graph of all contours of the facade image and to retrieve similar windows from a database of images of facades.

Index Terms—Attributed relational graph, kernel on graphs, window extraction, facade retrieval, inexact graph matching

1. INTRODUCTION

In the past few years, with the emergence of different geographical 3D browsers (Google Street View, Geoportail, Microsoft Live Earth), there has been renewed interest in cities modeling. Several approaches to model the urban environment have focused on a rough model (polyhedron walls, roof...). Recent approaches of research [1, 2, 3] try to analyze the real images of facades to provide rich information on buildings and to add realism for visualization. In this context, detection and classification of primitives (windows, doors, signs, cars ...) can improve navigation in these cities. Powerful object retrieval methods are based on local features such as Point of Interest (PoI) [4] or region-based descriptions [5]. Edge fragments appear to be relevant for architecture information on building facades. Regarding previous works [6, 7] which consider exclusively or mainly, contour fragments as the information support, the weakness of intrinsic information supported by edges requires to emphasize underlying structure of the objects in the description. Following the idea of

This work is supported by ANR (the french National Research Agency) within the scope of the iTOWNS research project (ANR 07-MDCO-007-03).

perceptual grouping of contours used by Ferrari et al. [8], we propose to design a kernel similarity function for structured sets of contours. Objects are then represented by fragments of contours with their own characteristics and by a relational graph on these fragments, where the vertices of the graph are contour segments extracted from the image and the edges of the graph represent their spatial relationships. This paper is organized as follows: First, we focus on similarity functions between objects represented by an attributed relational graph of segments of contours; to compare these graphs, we adapt kernels on graphs [9, 10] in order to define a kernel on paths more powerful than previous ones; then we extract a window as a sub-graph of the graph of all contours of the facade image and retrieve similar windows from a database of images of facades.

2. KERNEL ON GRAPH OF CONTOURS

2.1. Representation of architectural entities by Attributed Relational Graphs

Contours seem intuitively relevant to hold architecture information from building facades. In each image, edges are extracted, extended and polygonalized (figures 3 (b) (d)). In order to consider the set of edges as a whole, we represent it by an Attributed Relational Graph (ARG). Each line segment C_i is represented by a vertex v_i of this graph and the relative position of line segments $C_i C_j$ is represented by the edge e_{ij} of the graph. The topological information (such as parallelism, proximity) can be considered only for the nearest neighbors of each line segment. We use the Voronoi diagram to find the segments that are the closest to a given segment. An edge in the ARG represents the adjacency of two Voronoi cells that is to say the proximity of two line segments.

In order to be robust to scale changes, a segment is only characterized by its direction (horizontal or vertical). If Θ is the angle between line segment C_i and the horizontal axis ($\Theta \in [0, 180[$), C_i is represented by the vertex $v_i = (\cos(2\Theta), \sin(2\Theta))^T$.

The edge (of the graph) $e_{ij} = (v_i, v_j)$ represents the adjacency between line segments C_i and C_j . It is characterized by the relative positions of the centers of grav-

ity of segments C_i and C_j , denoted $g_{C_i}(Xg_{C_i}, Yg_{C_i})$ and $g_{C_j}(Xg_{C_j}, Yg_{C_j})$. The edge e_{ij} is then characterized by $e_{ij} = (Xg_{C_j} - Xg_{C_i}, Yg_{C_j} - Yg_{C_i})^T$.

2.2. Graph matching with Kernel

The problem of classifying architectural entities can be considered as a problem of inexact graph matching. The problem is twofold : first find a similarity measure between graphs of different sizes, then find the best match between graphs in an “acceptable” time. To address this last computational complexity problem, the best matching path is efficiently given by the “branch and bound” algorithm.

For the former issue, recent approaches of graph comparison [9], [10] propose to consider graphs as sets of paths. A path h in a graph $G = (V, E)$ is a sequence of vertices of V linked by edges of E : $h = (v_0, v_1, \dots, v_n)$, $v_i \in V$.

Kashima et al. proposed [10] to compare two graphs G and G' by comparing all possible paths of same length of both graphs. The problem of this kernel is its high computational complexity, acceptable with graphs of chemical molecules which have symbolic values, but unaffordable with our attributed graphs. Other kernels on graphs were proposed by Lebrun et al. [9], which are faster than Kashima kernel.

We design a new kernel based on [9]:

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

with h_{v_i} path starting from v_i .

K_C represents a kernel on paths. Several K_C were proposed [9] (sum, product...). We tested all these kernels and the best results were obtained with the following one, where e_j denotes edge (v_{j-1}, v_j) :

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

K_v and K_e are the minor kernels which define the vertex similarity and the edge similarity. We propose these minor kernels:

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

Our kernel aims at comparing sets of contours, from the point of view of their orientation and their relative positions. However, some paths may have a strong similarity but provide no structural information. For example, paths whose all vertices represent segments almost parallel. To deal with this problem, we can increase the length of paths, but the complexity of calculation becomes quickly unaffordable. Thus the solution we propose is to add a weight $O_{i,j}$ in the kernel K_C

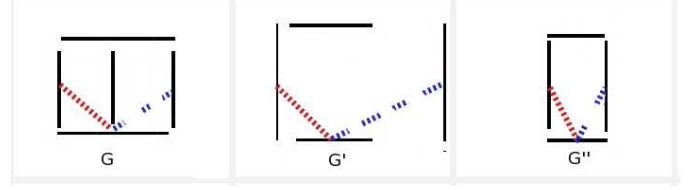


Fig. 1. Examples of graphs with a path in dot lines. The scale penalty will decrease the similarity of G' with G

which penalizes the paths whose segment orientations do not vary.

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

with ϕ_{ij} the angle between vertices i and j .

Moreover the perceptual grouping of sets of contours is crucial for the recognition. For example in figure 1, the question is: has the rightmost contour of graph G' to be clustered with the other contours to form an object or not? To model this information, we add a scale penalty Se_i .

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

Although $O_{i,j}$ and Se_i are illustrated in the case of graph of window contours, these weights remain relevant for other objects since no real perceptual information could be extracted from a set of only parallel contours.

Our final kernel K_C becomes ($Se_i \in [0, 1]$ and $O_{i,j} \in [0, 1]$):

$$K_C(h_{v_i}, h') = K_v(v_i, v'_0) + \sum_{j=1}^{|h|} Se_j O_{j,j-1} K_e(e_j, e'_j) K_v(v_j, v'_j) \quad (2)$$

3. CLASSIFICATION OF WINDOW CANDIDATES WITH PRIOR KNOWLEDGE

3.1. Detection of window candidates

Lee et Nevatia [11] proposed a profile projection method to extract windows by exploiting the geometric property of 2D rectangles and alignment of the building windows. We used this method to extract windows candidates, and obtained a database with 70 windows and 230 false detections (negative examples). We tested our method based on kernel on graph of contours to remove the false detections on this database of 300 images. Each image contains between 10 and 30 line segments. We tried paths of lengths between 3 and 10. With a 3-edges length, we do not fully take advantage of the structure of the graph, and with a 10-edges length, the time complexity becomes problematic. We set up : $|h| = 8$.

3.2. Active classification of window candidates

We first evaluate our kernel, the interest of weights proposed in this paper and compare with a method based on sets of features using point of interest (combination of MSER region detectors and SIFT descriptors [4]). We simulated an active learning scheme using Support Vector Machines classifiers, where the user annotates a few images at each iteration of relevance feedback, thanks to the interactive retrieval system [9]. Each retrieval session is initialized with one image containing an example of window. The system labels at each iteration one image either as window or as false detection, and the system updates the ranking of the database according to these new labels. The whole process is iterated 100 times with different initial images and the Mean Average Precision (MAP) is computed from all these sessions (figure 2).

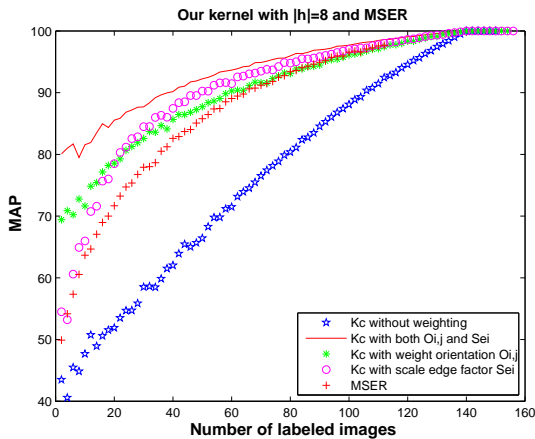


Fig. 2. Comparison of various kernels on paths with weighting by scale and/or orientation of the contours and the MSER method

We compared our kernels with and without the various weights proposed in section 2.2. With only one example of window and one negative example, we obtain 42 % of correct classification with the kernel without weighting. This percentage goes up to 54% with the scale weighting, to 69% with the orientation weighting, and to 80 % with both weightings. Results with weightings are much more improved after a few steps of relevance feedback than without weighting, to reach 90 % with 40 examples (20 positive and 20 negative), instead of 100 examples without weighting. The results with MSER are not as good as the results with kernel on graph on contours. The structure of the object is better taken into account with graph on contours than with MSER.

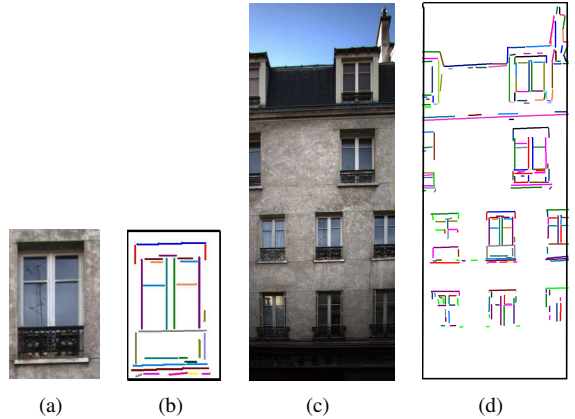


Fig. 3. Find the subgraph of window the most similar to graph of facade (a) Window query. (b) Graph of window (a). (c) Facade.(d) Graph of facade (b).

4. EXTRACTION OF WINDOWS WITHOUT PRIOR KNOWLEDGE

4.1. Detection of windows in facade

In section 2.2, we used kernels on graph of contours in order to classify graphs according to their similarities. In this section we want to avoid the preliminary stage (3.1) and we propose to exploit directly the graph of facade segments in order to detect window position. We want to find the subgraph of G' the most similar to graph G . In our case, G is the graph of a window and we look for windows within a facade (Fig.3).

In our kernel (eq. 1), for each vertex v_i of G we search for the best match between $h_{v_i} \in G$ (path starting from vertex v_i) and $h' \in G'$ (in this section, we set up $|h| = 8$). From this best path h' , we define a bounding box $B_{h_{v_i}, h'}$ around path h' as a region of interest (figure 4 (a)). In order to extract the windows, we assign a value to each pixel within this region of interest. All best paths h' , each corresponding to one h_{v_i} of G will thus vote for a set of pixels of G' . The value is the similarity of each h' . The result is an accumulation of similarities in several bounding boxes (figure 4 (b)). For each pixel (x,y) of image I we have:

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

where $B_{h_{v_i}, h'}(x, y) = K_C(h_{v_i}, h')$ if pixel (x,y) is in the bounding box and 0 otherwise.

In figure 4 (b), we note that the best paths are accumulated on two windows, but other windows are missed. To detect more windows, we need more paths. That is the reason why we propose for each vertex v_i to select the k best matches between h_{v_i} and h' . For each vertex v_i in graph G , we take not only the best match, but the k best matches with paths

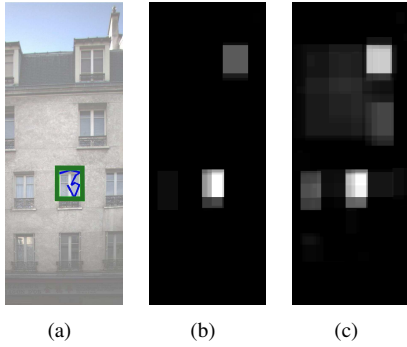


Fig. 4. Extraction of windows from one example (a) Best path and its bounding-box. (b) Accumulation of regions of interest with 1 path by vertex. (c) Accumulation of regions of interest with 10 paths by vertex

starting from vertex v_i . The result is :

$$I(x, y) = \sum_{i=1}^{|V|} \sum_{j=1}^k B_{h_{v_i}, h'}^j(x, y) \quad (4)$$

We extract thus more region of interest (Figure 4(c)).

4.2. Retrieving similar windows from a facade database

This kernel framework can also be used to retrieve images of facade containing similar windows. In the following experiments we took, as query, the graph of window of figure 3 (a). The database is composed of 200 urban images including 50 facades. Facades are ranked according to eq.(1). In eq.(4), we set up $k = 10$ which gives a good compromise between several true negative ($k = 1$) and several false detections.

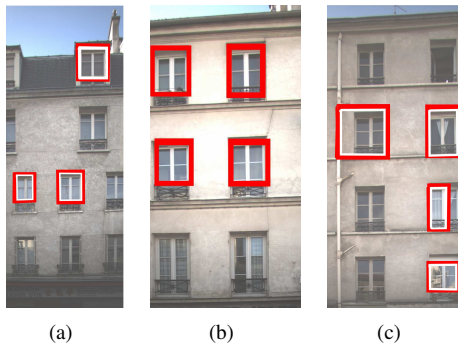


Fig. 5. Facade retrieval and extraction of windows Three first retrieved images. Window query belongs to facade (a) and in facades (b) (c) similar windows are found. Windows detected have the same structure (balconies and jamb).

First results of window extraction and retrieval are shown in figure 5. Among these best facades retrieved, we have with the rank 1 the facade containing the query window (5 (a))

and the facades ranked next contain some similar windows (5 (b)(c)). Windows detected have the same structure (balconies and jamb).

5. CONCLUSIONS

In this paper, we addressed the problem of inexact graph matching, window detection and facade retrieval by working out a new contour-based method using graph on contours.

We have also shown that objects can be represented by sets of contours. The new kernel we proposed is able to take into account orientations and proximity of contours in the structure. We are able to extract a window as a sub-graph of the graph of all contours of the image. Moreover we can retrieve images of facade containing similar windows.

To improve the extraction method, we will propose another voting system to segment the building and find more windows. A further step will also be to extend our kernel-based system to other more complex “urban objects”, cars or pedestrians.

6. REFERENCES

- [1] H. Mayer and S. Reznik, “Building facade interpretation from uncalibrated wide-baseline image sequences,” *PandRS*, vol. 61, no. 6, pp. 371–380, 2007.
- [2] P. Müller, G. Zeng, P. Wonka, and L. Van Gool, “Image-based procedural modeling of facades,” in *ACM SIGGRAPH*, 2007, vol. 26.
- [3] H. Ali, C. Seifert, N. Jindal, L. Paletta, and G. Paar, “Window detection in facades,” in *ICIAP*, 2007.
- [4] 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*, vol. 65, pp. 43 – 72, 2005.
- [5] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: image segmentation using expectation-maximization and its application to image querying,” *PAMI*, vol. 24, no. 8, pp. 1026–1038, 2002.
- [6] J. Shotton, A. Blake, and R. Cipolla, “Contour-based learning for object detection,” in *ICCV*, 2005, pp. 503–510.
- [7] A. Opelt, A. Pinz, and A. Zisserman, “A boundary-fragment-model for object detection,” in *ECCV*, 2006, pp. 575–588.
- [8] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, “Groups of adjacent contour segments for object detection,” *TPAMI*, vol. 30, no. 1, pp. 36–51, 2008.
- [9] J. Lebrun, S. Philipp-Foliguet, and P.-H. Gosselin, “Image retrieval with graph kernel on regions,” in *ICPR*, dec 2008.
- [10] H. Kashima and Y Tsuboi, “Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs,” in *ICML*, 2004.
- [11] S.C. Lee and Ramakant Nevatia, “Extraction and integration of window in a 3d building model from ground view image,” in *CVPR*, 2004, pp. 113–120.