

Extraction of Windows in Facade Using Kernel on Graph of Contours

Jean-Emmanuel Haugeard*, Sylvie Philipp-Foliguet,
Frédéric Precioso, and Justine Lebrun

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

Abstract. In the past few years, street-level geoviewers has become a very popular web-application. In this paper, we focus on a first urban concept which has been identified as useful for indexing then retrieving a building or a location in a city: the windows. The work can be divided into three successive processes: first, object detection, then object characterization, finally similarity function design (kernel design). Contours seem intuitively relevant to hold architecture information from building facades. We first provide a robust window detector for our unconstrained data, present some results and compare our method with the reference one. Then, we represent objects by fragments of contours and a relational graph on these contour fragments. We design a kernel similarity function for structured sets of contours which will take into account the variations of contour orientation inside the structure set as well as spatial proximity. One difficulty to evaluate the relevance of our approach is that there is no reference database available. We made, thus, our own dataset. The results are quite encouraging regarding what was expected and what provide methods the literature.

Keywords: Relational graph of segments, kernel on graphs, window extraction, inexact graph matching.

1 Introduction

Several companies, like Blue Dasher Technologies Inc., EveryScape Inc., Earthmine Inc., or GoogleTM provide their street-level pictures either to specific clients or as a new world wide web-service. However, none of these companies exploits the visual content, from the huge amount of data they are acquiring, to characterize semantic information and thus to enrich their system.

Among many approaches proposed to address object retrieval task, local features are commonly considered as the most relevant data description. Powerful object retrieval methods are based on local features such as Point of Interest (PoI) [1] or region-based descriptions [2]. Recent works consider not anymore a

* The images are acquired by the STEREOPOLIS mobile mapping system of IGN.
Copyright images: ©IGN for iTOWNS project.

single signature vector as object description but a set of local features. Several strategies are then possible, either consider these sets as unorganized (bags of features) or put some explicit structure on these sets of features. Efficient kernel functions have been designed to represent similarity between bags [3]. In [4], Gosselin et al. investigate the kernel framework on sets of features using sets of PoI. In [4], the same authors address multi-object retrieval with color-based regions as local descriptions. Based on the same region-based local features, Lebrun et al. [5] presented a method introducing a rigid structure in the data representation since they consider objects as graphs of regions. Then, they design dedicated kernel functions to efficiently compare graphs of regions.

Edge fragments appear to be relevant key-support for architecture information on building facades. However, a pixel set from a contour is not as informative as a pixel set from a region. Regarding previous works [6], [7] which consider exclusively or mainly, contour fragments as the information supports, this lack of intrinsic information requires to emphasize underlying structure of the objects in the description. Independently, Shotton et al. and Opelt et al. proposed several approaches to build contour fragment descriptors dedicated to a specific class of object. Basically, they learn a model of distribution of the contour fragments for a specific class of objects. Although, they can be more discriminative for the learned class, they are not robust to noisy contours found in real images. Indeed, to learn a class, they must select clean contours from segmentation masks. Ferrari et al. [11] use the properties of perceptual grouping of contours.

Following this same last idea, we propose to design a kernel similarity function for structured sets of contours. First, objects are represented by fragments of contours and a relational graph on these contour segments. The graph vertices are contour segments extracted from the image and characterized by their orientation to the horizontal axis. The graph edges represent the spatial relationships between contour segments.

This paper is organized as follows. First, we extract window candidate using the accumulation of gradients. We describe initial method and present our improvement on the automatic setting of the scale of extraction. Then, we focus on similarity functions between objects characterized by an attributed relational graph of segments of contours. To compare these graphs, we adapt kernels on graphs [8], [9] in order to define a kernel on paths more powerful than previous ones.

2 Extraction of Window Candidates

In this section, we explain the extraction of window candidates. We are inspired by the work of Lee et al. [10] that uses the properties of windows and facades and we propose a new algorithm.

2.1 Accumulation of Gradient

In 2004, Lee et al. [10] proposed a profile projection method to extract windows. They exploited both the fact that windows are horizontally and vertically aligned in the facade and that they have usually rectangular shape. Results are

good and accurate on a simple database, where walls are not textured, windows are regularly aligned and there is no occlusion nor shadows. In the context of old historical cities like Paris, images are much more complex: windows are not always aligned (figure 1a), textures are not uniform, there are illumination variations, there may be occlusions due to trees, cars, etc. Since they are organized in floors, windows are usually horizontally aligned. We propose thus to firstly find the floors and then to work on them separately to extract the windows, or at least rectangles which are candidates to be windows. Moreover we improve this method by completely automatizing the extraction of window candidates by determining the correct scale of analysis.

Floor and Window Candidate Detection

In order to find the floors, the vertical gradients are computed (figure 1b), and their norms are horizontally accumulated to form an horizontal histogram (figure 1c). High values of this histogram correspond more or less to window positions whereas low values correspond to wall (or roof). The histogram is thus threshold to its average value, the facade is so split into floors (figure 1d). The process is repeated in the other direction (horizontal gradients, vertical projection) separately for each floor, giving the window candidates.

Automatic Window Candidate Extraction

As we need an accurate set of edges to perform the recognition, we used the optimal operators of smoothing and derivation of Shen-Castan [12] (optimal in the Canny sense). The operators of Canny family depend on a parameter linked to the size of the filter (size of the Gaussian for Canny filter for example) or, which is equivalent, to the level of detail of the edge detection. We will denote β this parameter.

If the smoothing is too strong, some edges disappear (figure 2) whereas if the smoothing is too weak, there is too much noise (texture between windows). Thus, the number of extracted floors p_β depends on β , but it does not regularly evolve with β (cf. figure 2d). It passes by a plateau which usually constitutes a good compromise.

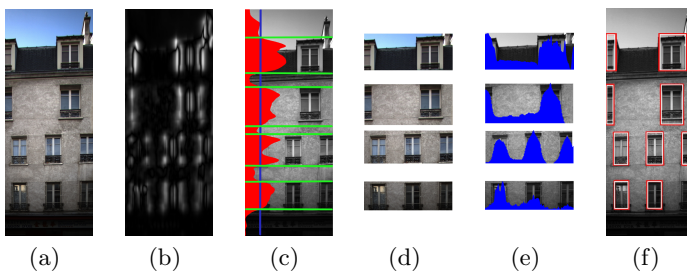


Fig. 1. Window candidate extraction. (a) Example of facade where the windows are not vertically aligned. (b) Vertical gradient norms. (c) Horizontal projection. (d) Split into 4 floors. (e) Vertical projection. (f) Window candidates.

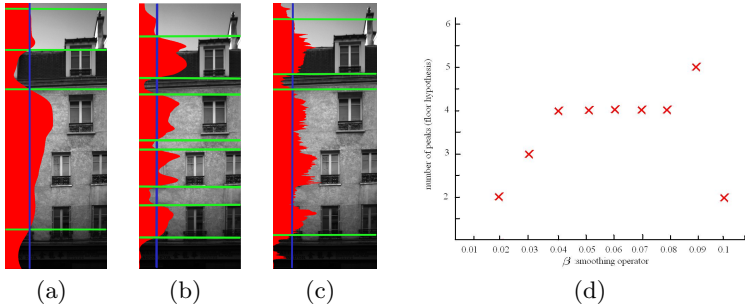


Fig. 2. The number of floors depends on the smoothing and derivation parameter β . (a) Strong smoothing. (b) Good compromise. (c) Weak smoothing. (d) Evolution of number of floors according to β .

In order to determine the value of β corresponding to this plateau, we compute a score S_{β_i} for each value β_i (β_i grows between 0 and 1). The idea is to maximize this score depending on the stability of the histogram and the amplitude of peaks H_{pj} .

$$S_{\beta_i} = \begin{cases} \frac{\overbrace{p_{\beta_{i-1}}}^{\text{Stability}}}{p_{\beta_i}} \cdot \frac{\overbrace{\sum_{j=1}^{p_{\beta_i}} \max H_{pj}}^{\text{average peak amplitude}}}{p_{\beta_i}} & \text{if } p_{\beta_{i-1}} < p_{\beta_i} \\ \frac{\sum_{j=1}^{p_{\beta_i}} \max H_{pj}}{p_{\beta_{i-1}}} & \text{else} \end{cases}$$

with p_{β_i} the number of peaks for β_i .

For each image, a value of β is evaluated to extract window candidates in each floor.

To summarize, the algorithm of window candidate extraction is:

Algorithm 1. Automatic Windows Extraction

Require: rectified facade image I_0

Initialization: $\beta_0 \leftarrow 0.02$

repeat

- 1) Compute vertical gradient norms
- 2) Project and accumulate horizontally these vertical gradient norms
- 3) Calculate evaluation score S_{β_i}
- 4) $\beta_i \leftarrow \beta_i + 0.01$

until $\beta_i = 0.3$

Choose $\beta_t = \operatorname{argmax}_{\beta_i} S_{\beta_i}$

Cut into floors with β_t according to the peaks. Compute the histogram of horizontal gradient norms on each floor with β_t and search the peaks of this vertical projection
 Rectangles are window candidates

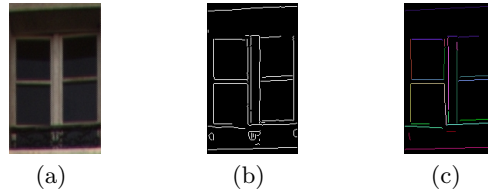


Fig. 3. Segmentation: the image is represented by a relational graph of line segment of contours. (a) Window candidate. (b) Edge extraction. (c) Polygonalization.

2.2 Representation of Window Candidates by Attributed Relational Graphs

After this first step, we have extracted rectangles which are candidates for defining windows. Of course, because of the complexity of the images, there are many mismatches and a step of classification is necessary to remove outliers. In each rectangle, edges are extracted, extended and polygonalized (figure 3). In order to consider the set of edges as a whole, we represent it by an Attributed Relational Graph (ARG). Each line segment is a vertex of this graph and the relative positions of line segments are represented by the edges of the graph. The topological information (such as parallelism, proximity) can be considered only for the nearest neighbors of each line segment. We use a 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 regions that is to say the proximity of two line segments.

In order to be robust to scale changes, a vertex 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 $v_i = \begin{pmatrix} \cos(2\theta) \\ \sin(2\theta) \end{pmatrix}$.

Edge (v_i, v_j) represents the adjacency between line segments C_i and C_j . It is characterized by the relative positions of the centres of gravity of C_i and C_j , denoted $g_{C_i}(Xg_{C_i}, Yg_{C_i})$ and $g_{C_j}(Xg_{C_j}, Yg_{C_j})$. Edge (v_i, v_j) is then characterized by $e_{ij} = \begin{pmatrix} Xg_{C_j} - Xg_{C_i} \\ Yg_{C_j} - Yg_{C_i} \end{pmatrix}$.

3 Classification and Graph Matching with Kernel

In order to classify the window candidates into true windows and false positives, we chose to use machine learning techniques. Support Vector Machines (SVM) are state-of-the-art large margin classifiers which have demonstrated remarkable performances in image retrieval, when associated with adequate kernel functions.

The problem of classifying our candidates can be considered as a problem of inexact graph matching. The problem is twofold : first, find a similarity measure

between graphs of different sizes and second, find the best match between graphs in an “acceptable” time. For the second problem, we opted for the “branch and bound” algorithm, which is more suitable with kernels involving “max” [5]. For the first problem, recent approaches propose to consider graphs as sets of paths [8], [9].

3.1 Graph Matching

Recent approaches of graph comparison 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 [9] to compare two graphs G and G' by a kernel comparing all possible paths of same length of both graphs.

The problem of this kernel is its high computational complexity. If this is acceptable with graphs of chemical molecules, which have symbolic values, it is unaffordable with our attributed graphs.

Other kernels on graphs were proposed by Lebrun et al. [5], which are faster than Kashima kernel:

$$K_{Lebrun}(G, G') = \frac{1}{|V|} \sum_{i=1}^{|V|} \max K_C(h_{v_i}, h'_{s(v_i)}) + \frac{1}{|V'|} \sum_{i=1}^{|V'|} \max K_C(h_{s(v'_i)}, h'_{v'_i}).$$

with $\begin{cases} h_{v_i} \text{ is a path of } G \text{ whose first vertex is } v_i \\ h'_{s(v_i)} \text{ is a path of } G' \text{ whose first vertex is the most similar to } v_i \end{cases}$

Each vertex v_i is the starting point of one path and this path is matched with a path starting with the vertex $s(v_i)$ of G' the most similar to v_i . This property is interesting for graphs of regions because regions carry a lot of information, but in our case of graphs of line segments, the information is more included in the structure of the graph (the edges) than in the vertices.

We propose a new kernel that removes this constraint of start (h_{v_i} path starting from v_i):

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

Concerning the kernels on paths, several K_C were proposed [5] (sum, product...). We tested all these kernels and the best results were obtained with this 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:

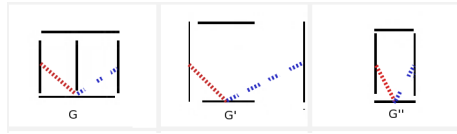


Fig. 4. Example: structures and scale edge problem. Is the segment of contour on the right in graph G' a contour of the object or not?

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

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 segment almost parallel. To deal with this problem, we can increase the length of paths, but the complexity of calculation becomes quickly unaffordable. To overcome this problem, we add in K_C a weight $O_{i,j}$ that 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 4, graphs G' and G'' have almost the same structure as graph G , but the rightmost contour is further away in graph G' than in the two other graphs. The question is: has this contour to be clustered with the other contours to form an object or not? To model this information, we add a scale factor Se_i .

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

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

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

4 Experiments and Discussions

In this section, we first compare our algorithm of window extraction to Lee algorithm [10]. Then we evaluate our kernel and the interest of the weights proposed in this paper.

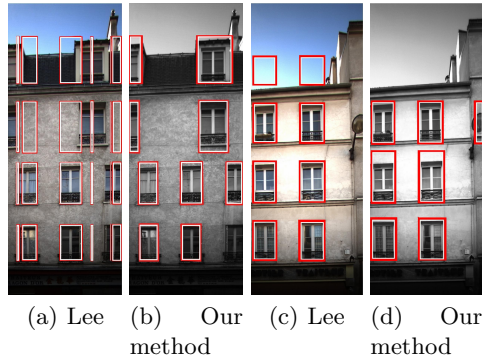


Fig. 5. Comparison of Lee and our method on complex cases. (a) (b) windows are not vertically aligned. (c) (d) chimneys induce false detection with Lee.

4.1 Window Candidate Extraction

Institut Geographique National (IGN) is currently initiating a data acquisition of Paris facades. The aim of our work is to extract and recognize objects present in the images (cars, windows, doors, pedestrians ...) of this large database. We have tested our algorithm on Paris facade database and compared it with Lee and Nevatia algorithm [10] (we denote it Lee in the figures). Images are rectified before processing. On simple cases we get results similar to Lee, but in more complex cases, when windows are not exactly aligned or when there is noise due to chimneys, drainpipes, etc (figures 5 and 6), we obtain better results. Moreover, our algorithm is automatic, it chooses by itself the correct scale of analysis to properly extract the contours.

4.2 Classification of Window Candidates

We tested our method to remove the false detections on a database of 300 images, for which we had the ground-truth : 70 windows and 230 false detections

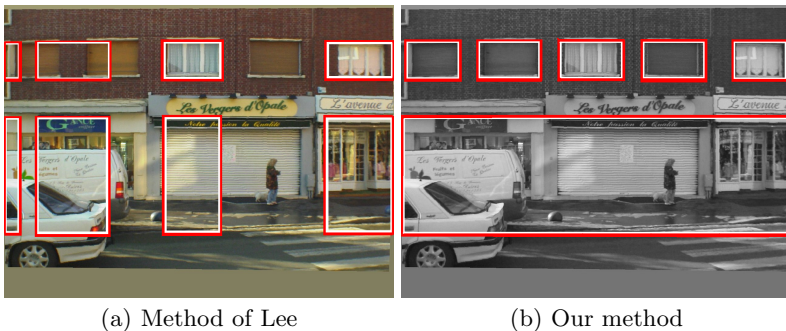


Fig. 6. Comparison of Lee and our method on a complex case: windows are not exactly horizontally aligned and there is a lot of distractors

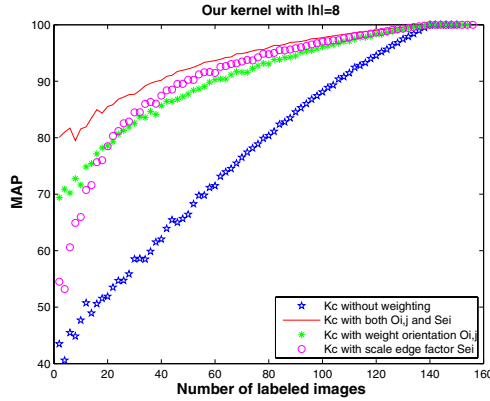


Fig. 7. Comparison of versions kernels on paths with weighting by scale and orientation of the contours

(negative examples). Each image contains between 10 and 30 line segments. We tried paths of lengths between 3 and 10. With a 3-length edges, we do not fully take advantage of the structure of the graph, and with a 10-length edges, the time complexity becomes problematic. We opted for a compromise : $|h| = 8$.

Each retrieval session is initialized with one image containing an example of window. We simulated an active learning scheme, where the user annotates a few images at each iteration of relevance feedback, thanks to the interface (cf. Fig. 8). The system labels at each iteration one image as window or false detection, and the system updates the ranking of the database according to these new labels. The

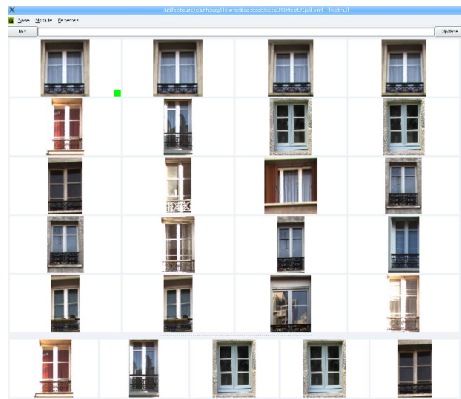


Fig. 8. The RETIN graphic user interface. Top part: query (left top image with a green square) and retrieved images. Bottom part: images selected by the active learner. We note that the system returns windows and particularly windows which are in the same facade or have the save structure than the query (balconies and jambs).

whole process is iterated 100 times with different initial images and the Mean Average Precision (MAP) is computed from all these sessions (figure 7).

We compared our kernels with and without the various weights proposed in section 3. 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. Figure 8 shows that we are also able to discriminate between various types of window, the most similar being the windows of the same facade or of the same number of jambs.

5 Conclusions

We have proposed an accurate detection of contours from images of facades. Its main interest, apart the accuracy of detection is that it is automatic, since it adapts its parameter to the correct scale smoothing of analysis. We have also shown that objects extracted from images can be represented by a structured set of contours. The new kernel we have proposed is able to take into account orientations and proximity of contours in the structure. With this kernel, the system retrieves the most similar windows from facade database. The next step is to free oneself from the step of window candidate extraction, and to be able to recognize a window as a sub-graph of the graph of all contours of the image. This process involving perceptual grouping will then be extended to another type of objects like cars for example.

Acknowledgments. This work is supported by ANR (the french National Research Agency) within the scope of the iTOWNS research project (ANR 07-MDCO-007-03). Copyright images: ©IGN for iTOWNS project.

References

1. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. *International Journal of Computer Vision* (2005)
2. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004)
3. Shawe-Taylor, J., Cristianini, N.: *Kernel methods for Pattern Analysis*. Cambridge University Press, Cambridge (2004)
4. Gosselin, P.-H., Cord, M., Philipp-Foliguet, S.: Kernel on Bags for multi-object database retrieval. In: *ACM International Conference on Image and Video Retrieval*, pp. 226–231 (2007)
5. Lebrun, J., Philipp-Foliguet, S., Gosselin, P.-H.: Image retrieval with graph kernel on regions. In: *IEEE International Conference on Pattern Recognition* (2008)

6. Shotton, J., Blake, A., Cipolla, R.: Contour-Based Learning for Object Detection. In: 10th IEEE International Conference on Computer Vision (2005)
7. Opelt, A., Pinz, A., Zisserman, A.: A Boundary-Fragment-Model for Object Detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 575–588. Springer, Heidelberg (2006)
8. Suard, F., Rakotomamonjy, A., Bensrhair, A.: Détection de piétons par stéréovision et noyaux de graphes. In: 20th Groupe de Recherche et d'Etudes du Traitement du Signal et des Images (2005)
9. Kashima, H., Tsuboi, Y.: Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In: International Conference on Machine Learning (2004)
10. Lee, S.C., Nevatia, R.: Extraction and Integration of Window in a 3D Building Model from Ground View Image. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004)
11. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of Adjacent Contour Segments for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008)
12. Shen, J., Castan, S.: An Optimal Linear Operator for Step Edge Detection. *Graphical Models and Image Processing* (1992)