

Contextual features and structural description for facade detection

Antoine Fond, Marie-Odile Berger and Gilles Simon¹

Abstract—In that article we focus on facade detection in order to improve image/model buildings matching for pose computation in urbain environnement. We use a two-step design. First a cascade of LogitBoost classifiers using features which describe local context selects a few windows from a set of windows drawn on an *a priori* distribution. These facade candidates are then more internally described using their Haar-Fourier representation. Eventually they are discarded or kept by a strong SVM classifier. Results are computed from a 410-set of urban images.

I. INTRODUCTION

Although facade detection and recognition has interesting potential uses in areas such as autonomous navigation and augmented reality, these tasks face a number of difficulties, such as perspective deformations, lighting changes, repeated patterns and occlusions. In [12], we proposed a fast and efficient method to automatically detect orthogonal vanishing points in images of urban environments. From this knowledge, it is straightforward to orthorectify facades aligned with the vanishing points. This procedure is likely to facilitate tasks such as pose computation and localization [2]. In this paper, we go one step further by delimiting areas corresponding to the facades, which will be helpful in tasks such as image indexation, model-to-image matching and building recognition.

In the spirit [5], [13], our method is based on a two-step process. In the first step, a cascade of classifiers is used to extract a limited number of candidate windows. Several “facadeness” measures based on contextual information are evaluated for a set of overlapping windows of different sizes. In order to reduce the computational complexity, these windows are drawn from an *a priori* distribution learned from a basis of examples. In the second step, a Support Vector Machine (SVM) is used to prune the remaining candidates. This procedure is based on a more structural descriptor that combines Haar-wavelet and Fourier transforms. We built a dataset of 920 images of urban scenes coming from the subsets “street” and “building” of ImageNet. All these images have been orthorectified according to the dominant facade and we manually labeled the facades, leading to 1324 facades for the whole dataset. 510 images were used to learn the parameters of the method and 410 for the tests.

A. State of the Art

Two categories of methods have been proposed in the past to detect facades in an image. Geometric-based methods try to find rectangles whose edges are consistent with three

orthogonal vanishing points [8], [11], [4]. Statistic-based methods use different kind of features to identify regions that have a high probability of belonging to a facade [6], [2].

In [8], line segments are automatically detected and intersected to generate hypotheses of rectangles in agreement with the vanishing points. For each hypothesis, the input image is orthorectified and a histogram of gradient (HOG) is computed inside the warped rectangle. Hypotheses whose HOG contains more than two dominant horizontal and vertical directions are discarded. This method is computationally expensive generating many superfluous hypotheses. To keep the problem tractable and efficient, Mikusik et al. formulate the detection of the rectangles on a restricted neighborhood structure given by Delaunay triangulation [11]. The problem is then expressed as a search for the maximum a posteriori probability solution of a Markov random field. In [4], right corners are detected in the orthorectified image using a SVM. A Delaunay triangulation is performed from the right corners and a min-cut-like algorithm is used to generate windows in which a high density of right corners is observed. These methods allow to detect rectangular structures on a facade (windows, rows of windows, ...) but not (in general) an entire facade.

Other methods aim at segmenting the image into labeled regions, using supervised classification. Hoiem et al. pioneered these works by labeling constellations of superpixels into coarse categories: “ground”, “sky”, and “vertical” [6]. Likelihood functions are estimated using the logistic regression version of Adaboost. Weak learners are based on different kinds of features such as color, texture, location, shape and geometry. In this method, the label “vertical” can be attributed to any vertical object. A procedure dedicated to the extraction of facades is considered in [2]. A pixel-wise segmentation of the image is performed, by applying a classifier to each image patch of a given size to assign a class label to the center location of the patch. The segmentation uses a multi-class SVM based on Integral Channel Features. By applying the classifier exhaustively, a probability estimate is obtained for each image pixel over five classes: “facade”, “sky”, “roof”, “vegetation” and “ground”. The main assumption of this approach is that local patches contain enough information to decide between these classes. However, a coarser granularity level has to be used for segmentation if one want to consider more global features in facade characterization.

Our method is both geometric (we consider orthorectified images and try to extract rectangles fitting the borders of the buildings) and statistic (supervised learning based on

¹Université de Lorraine, INRIA, LORIA, 615 Rue du Jardin botanique, 54506 Vandoeuvre-Is-Nancy, France antoine.fond@loria.fr

different features is used to identify such rectangles). It is close to that described in [1], which characterizes an object in its surrounding environment using several features (color contrast, edge density, ...) computed in sliding windows. A Naive Bayes approach combining these features allows to select a limited number of candidate objects. In the spirit of the “objectness” measure proposed by Alexe et al., we define in this paper a “facadeness” measure. However, although some features used by Alexe et al. are reused in this paper, some others, like their global saliency measure, are not suitable to our context.

The cascade classification procedure as well as the facadeness measures we use to select the candidate windows are presented in section II. The second step of the algorithm and particularly the Haar-Fourier descriptor are described in section III. Finally, our method is assessed in section IV.

II. SELECTION OF CANDIDATE WINDOWS

A. *A priori* windows distribution

A lot of object detection methods use sliding-windows. Sliding windows can be seen as a discrete set of the 4d space (x, y, l, h) on a regular grid. If the rectangular shape of the windows is a good point for seeking rectified facades, some of selection metrics use the contour of the facade. So we need that windows tightly fit the facade. Such a tight fitting means a small grid of the 4d space of windows. As well as for learning as for testing we define the following criteria $s_{pascal} = \frac{w_d \cap w_{gt}}{w_d \cup w_{gt}}$ in order to measure the fitting of the window compared to the ground truth w_{gt} . To get a chance of fitting all the ground truth windows with a score $s_{pascal} \geq 0.8$ we need to subdivide the 4d space on a 10^7 regular grid. In practice only a few of these windows can be a good facade candidate. Indeed, architectural and shooting constraints limit the shape of the windows (aspect ratio) and their position in the image. For instance, it is extremely not likely to find a long facade on the right-down corner of the image.

Our strategy is to learn that probability law $p(W)$, $W \in \mathbb{R}^4$ on a set of samples and draw windows from it. We choose a gaussian kernel density estimator to estimate $p(W)$ (Fig. 1). The variance of the gaussians is set to minimize the number of samples we need to get 90% chances of finding a “good” window among them for all the images of the database. We make vary the variance $\Sigma^2 = diag(\sigma_x^2, \sigma_y^2, \sigma_h^2, \sigma_l^2)$ and for each value of the variance we estimate $p(W)$. Then we draw $(10, 10^2, \dots, 10^7)$ samples for each images of the database. For each of those sampling we compute the number of “good” windows. We count only one good window for every ground truth window. We then find the sampling for which the ratio of the number of “good” windows to the number of ground truth windows is more than 90% (Fig. 2).

B. Measures of facadeness

We can limit the number of candidate windows to 100000 using $p(W)$, which means a ratio of “good” windows of about 80% on the whole learning database as well as on the testing database. However, it is still a very high number for

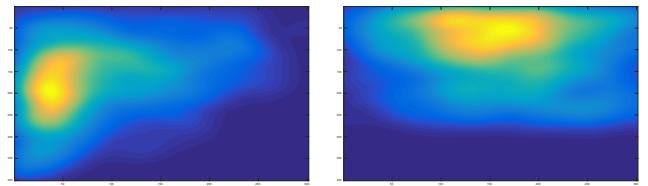


Fig. 1. Marginals of the *a priori* distribution $p(W)$ in (h, l) (left) and (x, y) (right).

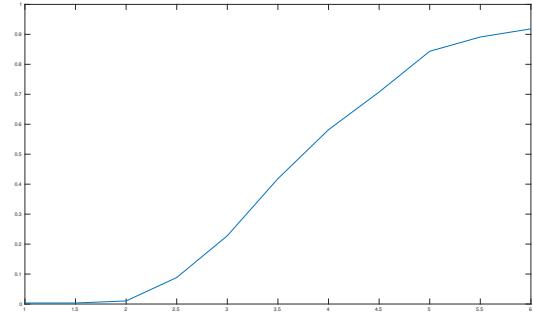


Fig. 2. Ratio of good windows over the learning database in function of the number of draws, expressed in logarithmic scale.

evaluating a strong classifier on each window. Some ad hoc features can be used to discard most of the windows and keep the best candidates. As these features have to be evaluated on a large set of windows, they have to be computed very fast (in a constant time regarding the size of the windows).

These features rely on three hypotheses : a facade is rectangular, a facade is structured by rectangular parts and a facade is different from its local context through its structure and its color. We use the notation $w = W(x, y, l, h)$ for the rectangle at position (x, y) with width l and height h . We note $r = R(I, w)$ the region of the image inside w .

As a rectified facade is rectangular, we expect a strong contour along the edges of the rectangle w . We thus define the contour feature:

$$\begin{aligned} s_{cp}(w) &= \frac{1}{2(l+h)} (\sum R(E_x, W(x-\alpha, y, \alpha, h)) \\ &\quad + \sum R(E_x, W(x+l+\alpha, y, \alpha, h)) \\ &\quad + \sum R(E_y, W(x, y-\alpha, l, \alpha)) \\ &\quad + \sum R(E_y, W(x, y+h+\alpha, l, \alpha))), \end{aligned} \quad (1)$$

where α is the thickness of the band around w . E_x and E_y are the images of the horizontal and vertical Dollar’s contour, respectively. Figure 3 shows the values of s_{cp} at each position of an example image. The feature s_{cp} is computed in constant time using the integral image of Canny edge detector keeping only horizontal and vertical contours. The use of integral image for sum of region is detailed in [14].

A rectified facade is mainly made of rectangular architectural features (windows, balcony, door, bricks, ...) and



Fig. 3. Values of s_{cp} computed at each position (x, y) of a window having the same size as the ground truth (in green). The position corresponds to the top left corner of the window.

uniform area (painting, rendering, ...). These rectangular features are present at different scale (the building itself, the floors, the windows, the bricks, ...). Thus the Haar wavelet transform DWT_{haar} of a facade, which is a multi-scale representation on a rectangular basis functions, is very sparse. The sparse dictionary using SPAMS [10] is composed of functions really close to Haar functions confirming that guess. We measure the facade structure of a region by the entropy of its wavelet coefficients. We define the following feature :

$$s_{he}(w) = H_S(DWT_{haar}(R(I)), w), \quad (2)$$

where $H_S(x) = -\sum_i \frac{|x_i|}{\|x\|_1} \log(\frac{|x_i|}{\|x\|_1})$ is the Shannon entropy (Fig. 4). The feature s_{he} is computed using integral images of $|x| \log(|x|)$ and $|x|$ from the sub-bands of the wavelet transform of I.



Fig. 4. Values of s_{he} computed at each position (x, y) of a window having the same size as the ground truth (in green). The position corresponds to the top left corner of the window.

Moreover, the more a facade appears differently from its surrounding environnement the more we can make it out. If its local context is made of unrectified building, or complex structures such as trees, bushes, cloudy sky, the facade appears more obviously. Thus the wavelet coefficients distribution of a thick ring around the facade is different from the distribution inside the facade. We estimate these distributions using histograms over the wavelets coefficients. These histograms can be compared using a χ^2 test. We define

$$s_{hd}(w) = \sqrt{\sum_i \frac{(h_{+\beta,i}^H - h_i^H)^2}{h_i^H}}, \quad (3)$$

where $h_{+\beta}^H$ and h^H are the histograms over the wavelet coefficients from the image regions $R(I, w_{+\beta})$ and $R(I, w)$

(resp.) with $w_{+\beta} = W(x - l\beta, y - h\beta, l(1 + 2\beta), h(1 + 2\beta))$ the rectangle enlarged by a factor β in all directions (Fig. 5). The wavelets transform is quantized into 10 bins. Using one integral image per bin allows us to compute the histogram in constant time (4×10 operations).

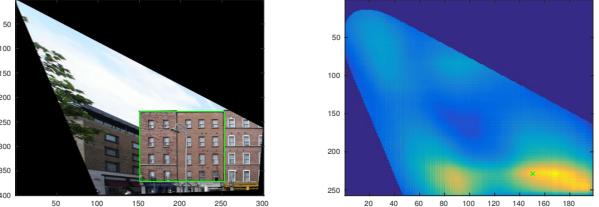


Fig. 5. Values of s_{hd} computed at each position (x, y) of a window having the same size as the ground truth (in green). The position corresponds to the top left corner of the window.

Eventually a facade is different from its environnement by the difference of color distribution. This color contrast can discard a facade from the sky or a grey facade from a red adjacent one. A χ^2 test is used to compare the color distribution. We define :

$$s_{cc}(w) = \sqrt{\sum_i \frac{(h_{+\gamma,i}^C - h_i^C)^2}{h_i^C}}, \quad (4)$$

where $h_{+\gamma}^C$ and h^C are the color histograms for the image regions $R(I, w_{+\gamma})$ and $R(I, w)$ (resp.), with $w_{+\gamma}$ the rectangle enlarged by a factor γ in all directions (Fig. 6). The LAB color space is quantized into $256 = 4 \times 8 \times 8$ bins. The histogram is computed in 4×256 operation using one integral image per bin.

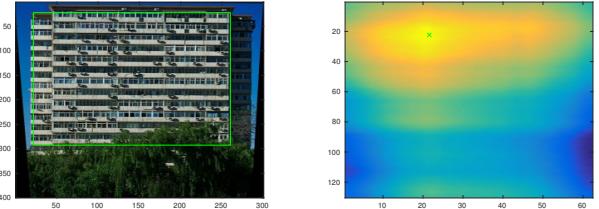


Fig. 6. Values of s_{cc} computed at each position (x, y) of a window having the same size as the ground truth (in green). The position corresponds to the top left corner of the window.

The different features are parametrized by constants α, β, γ . These parameters are learned so as to maximize the separability of negatives samples and positives samples. We compute the conditional probability of the values of each feature knowing whether the window is positive of negative. This feature likelihood is computed using the histogram of positive windows (“good” windows regarding the ground truth) and negative windows (what remains from the initial 10000-drawing) for all the images of the learning database. We set the parameter to the value that separate the most these histograms. The separability measurement is the previously

defined criteria s_{pascal} for the area under the curves. We get $\alpha = 8$, $\beta = 0.3$ and $\gamma = 0.24$ (Fig. 7).

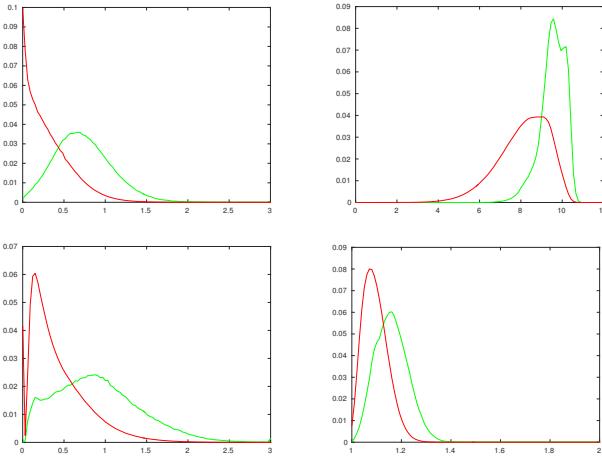


Fig. 7. Likelihood of the features s_{cp} (top left), s_{he} (top right), s_{hd} (bottom left) and s_{cc} (bottom right). The likelihood of the positive (negative) windows is shown in green (red).

C. Cascade of LogitBoost classifiers

If all the features are computed in constant time regarding the size of the windows, these constants can be very different from one another. Thus the color contrast s_{cc} needs more than 2^{10} operations whereas s_{cp} needs only 2^4 . That is why we chose a cascade classification scheme. The goal of such a scheme is to classify first with the less costly features. The most costly features are then computed only on the survivors of the previous stages of the cascade classification. The classifiers of each stages are here LogitBoost classifiers C which combine linearly weak stump classifiers $c_{i,j}$ [7]:

$$C(x) = \sum_{i=0}^N w_i c_{i,j}(x) \text{ with } c_{i,j}(x) = \begin{cases} 1 & \text{if } x_j \geq \xi_{i,j} \\ -1 & \text{else} \end{cases} \quad (5)$$

The cascade is made of 3 stages. The learning step of each LogitBoost classifier of each stage seeks to maximize the true positive ratio (facade classified as facade) as well as to favor a high reject ratio from one stage to another. We favor positive samples over negative samples by weighting them. Doing so the learning of LogitBoost artificially tends to behave as expected. The first stage uses s_{he} and s_{cp} with a LogitBoost classifier with 16 weak classifiers. The second stage uses s_{he}, s_{cp} and s_{hd} with 32 weak classifiers and finally the last stage combines all the features s_{he}, s_{cp}, s_{hs} and s_{cc} with 64 weak classifiers. The results of this cascade is about 90% of rejected windows regarding the 100000 initial drawing and 85% of true positives.

Among the 10000 windows classified as positive that remains at the end of the cascade, a lot of them overlap each other. The non-uniformity of $p(W)$ strengthens that effect. One last step is to only select a subset of the surviving windows which have a strong classification score and which

do not overlap each other. We use the same greedy approach of this problem that in the ‘objectness’ [1]. It consists in taking the best scored window and then to find the second best which does not overlap the first one and so on. Actually the ‘facadeness’ score is not so reliable for such a greedy approach. That is why we authorize overlaps until $s_{pascal} < 80\%$. That corresponds to the limit above which we consider two windows to be identical. In practice, about 400 facade candidates remain after that step.

III. FACADE CLASSIFICATION

A. Haar-Fourier descriptors

The modeling of a facade in the previous selection step is weak. It does not define sufficiently a facade and some ambiguities remain. In the surviving facades at the end of the cascade we can find large uniform area surrounded by complex details. It can be for example sky area between buildings or road area between cars. In these cases all the features of the facadeness have a high value. A more structural description of a facade is needed to discard these cases. We propose here a refined modeling that describes the internal structure of a facade. We still make the hypothesis that a facade is made of rectangular parts at different scales. These parts are often repeated over the horizontal or vertical directions. These two observations lead us to build a facade descriptor using the Fourier transform \mathcal{F} of the modulus of each sub-band (at scale j and direction d) of a Haar wavelet transform:

$$D = (|\mathcal{F}(|I * W_{j,d}|)|^2)_{j \leq J, d \leq 3} \quad (6)$$

Repetitive patterns at different scales and at different locations on the facade (floor, windows, ...) cause the direct use of the global Fourier transform to fail. The wavelet transform decomposes the signal into several scales and directions (Fig. 8). Such a decomposition enables the frequency analysis to be more local and more robust to noisy frequency such as non rectified parts or occlusions. However, the transform tends to spread the spectrum of each sub-band, which makes it more sensitive to small deformations (windows shift for example). Indeed we can consider the case of an ideal facade as a pulse train. The wavelet transform tends to transform it into a Diracs comb. To rescale the spectrum to low frequencies we take the modulus of the wavelet transform as it is done in the wavelet scattering [3].

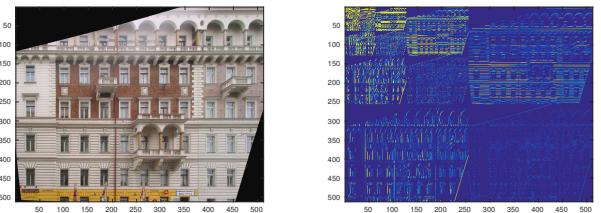


Fig. 8. Haar wavelet transform of an image of a facade.

In practice the repetitions mostly occurs on the vertical and horizontal directions. So as to decrease the size of the descriptor D we remove the high frequency coefficients of the diagonals (Fig. 9). The descriptor is computed in resizing the windows to 128×128 . Only the coefficients inside the smallest inscribing square are kept (Fig. 9). D is then a 8192 dimensional vector. A maxpooling step [9] reduces the dimensionality to 1280. The size of the patches are divided by two from one scale to another, starting from 4×4 for the highest resolution.

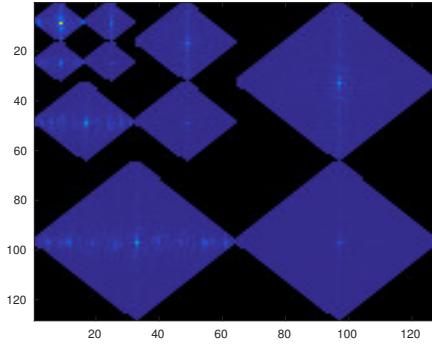


Fig. 9. Haar-Fourier descriptor for the image in Fig. 8.

B. SVM classification

We exploit these structural description of a facade in the last step of our facade detection method. The binary classification between “facade” / “non-facade” uses a linear SVM on the descriptor $D(w)$ of the surviving windows of that cascade. The description of a facade by D does not use the context but only the internal structure. Thus, rather than using the ground truth of our contextual database, the SVM is learned using a second database of 2361 facades labeled positive in images coming from several public databases used in image segmentation. The negative samples come from images of urban context from ImageNet (trees, streets, cars, signals, sky, ...) for a total of 5540 images. That database is divided into 1574 positive samples and 3650 negative samples used for learning. The remaining images are used for test purpose. The ratio of correct classification is 98% on the learning set and 95% on the testing set.

C. Ranking from overlapping map

In many cases it is not possible to differentiate a sub-part of a facade from a full facade without context information. If we remove one floor of a facade it is still a facade. So most of the sub-part are classified as “facade” by the SVM. Eventually we exploit that observation to rank the resulting windows of the SVM. To do so we build a map M_o that represents the overlapping of every sub-part of facades (Fig. 12). For every pixel of that map we compute the sum weighted by the facadeness score of all the windows classified “facade” overlapping it. We define for a window $w = W(x, y, l, h)$ the integrated score of the map M_o over w :

$$s_{M_o}(w) = \sum R(M_o, w) \quad (7)$$

As a facade is made of many sub-parts of a facade, $s_{M_o}(w)$ of a promising facade candidate w must be high. Moreover $s_{M_o}(w)$ has to decrease quickly if we change the window w a little (moving its position as well as changing its shape). This behavior can be seen as high value of the laplacian $\Delta s_{M_o}(w)$. In order to rank the windows after the SVM classification we compute the 4d discrete laplacian $(x', y', l', h') \mapsto \Delta s_{M_o}(W(x', y', l', h'))$ for each window $w = W(x, y, l, h)$. These windows are sorted in descending order regarding the laplacian $\Delta s_{M_o}(w)$.

IV. RESULTS

At the end of the “facadeness” step we keep in average 387 windows which are potentially facades (with a standard deviation of 218). In order to evaluate this step we count the number of “good” windows (s_{pascal} with a ground truth window ≥ 0.8) among the facade candidates (with no more than one “good” window for each ground-truth window). We do so for every image of the learning database. At the end of the cascade, the ratio of good windows over the total number of ground truth windows is 75.0% for the whole database. This result has to be compared to the ratio of 80.3% of good windows at the beginning of the cascade. In other words, for a given input image, each ground truth window has 80.3% chance to be found among the 100000 initial candidates, and 75.0% to be found among the 387 candidates that remain after the cascade. These results are well generalized on the testing database with very close good windows ratio of 80.1% before and 74.3% after the cascade. Fig. 11 shows the selection results obtained after each step of the algorithm.

The purpose of the Haar-Fourier/SVM classification is to discard the still remaining false detection using a structural description. After this step the number of windows is reduced to an average of 210 (about 1/2) with a standard deviation of 157. The ratio of good windows is of 71.2% on the whole learning database and of 70.0% on the testing database. The factor of 1/2 can be explained by the still high number of sub-part of facade overlapping each other (the non overlapping step still authorize $s_{pascal} < 0.8$) which is exploited in the final ranking step. The ranking shows very good visual results with usually the main facade ranked below 5 (Fig. 12). More statistically the median rank of the ground-truth found windows is 36 (Fig. 10).

If the ratio of 71% could be considered low it has to be compared with human ability to select facades in images. For 3 manual selection of the learning database made by 3 different people, we got an average fitting ratio of 40% between all the combination of ground-truth sets of windows. The fitting threshold $s_{pascal} \geq 0.8$ is not to be blamed. The variability is caused by the difficulty of choosing how to segment one building into several facades and to exhaustively select them in the image.

The mean time of computation of the facadeness is of 2.2s per image for a Matlab code running on a Intel Xeon W3565

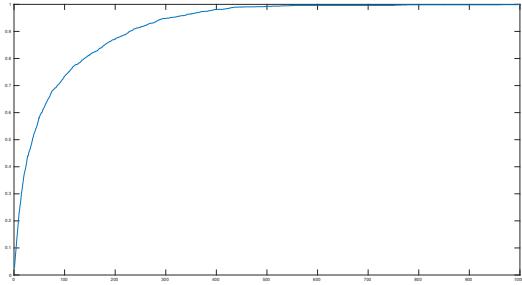


Fig. 10. Normalize cumulative histogram of the rank of the found ground-truth windows.

step	0	1	2	3	4	5	6
n	10^7	10^5	5.1×10^4	3.1×10^4	1.1×10^4	387	210
τ	1	0.80	0.79	0.76	0.74	0.74	0.71

Fig. 11. Mean number n of selected windows and ratio τ of ground truth windows remaining after each step i of the algorithm. Step 0 : regular grid; step 1: drawing from $p(W)$; step 2: first stage of the cascade; step 3: second stage of the cascade; step 4: third stage of the cascade; step 5: non-overlapping; step 6: SVM classification.

Quadcore 3.2 Ghz with 64 Go RAM. The fast computation of the features of facadeness is implemented in C and only called by Matlab. The full time of the method with the SVM on about 400 Haar-Fourier descriptors is about 4.5s.

V. CONCLUSION

In this paper, we proposed a fast method to generate facade hypotheses in images of urban environments. At the end of the process, a few hundreds of sorted windows are selected among which facades regions have a high probability to be found. Moreover, the overlapping of these windows highlights areas that visually correspond to expected results. In our future works we will examine how these windows and maps can be exploited to improve model / image matching and facade recognition processes. We will also continue to investigate the Haar-Fourier descriptor and determine to what extend this descriptor can be used as a global descriptor for image indexation.

REFERENCES

- [1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.
- [2] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. Instant outdoor localization and SLAM initialization from 2.5D maps. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1309–1318, Nov 2015.
- [3] J. Bruna and S. Mallat. Invariant scattering convolution networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1872–1886, 2013.
- [4] A. Fond, M.-O. Berger, and G. Simon. Prior-based facade rectification for AR in urban environment. In *ISMAR workshop on Urban Augmented Reality*, Fukuoka, Japan, September 2015.
- [5] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *ICCV 2009 - 12th International Conference on Computer Vision*, pages 237–244, Kyoto, Japan, 2009.
- [6] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584, NY, USA, 2005.
- [7] T. Hastie J. Friedman and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.
- [8] J. Košecká and W. Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):274 – 293, 2005.
- [9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
- [10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, 2009.
- [11] B. Mičušk, H. Wildenauer, and J. Košecká. Detection and matching of rectilinear structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008.
- [12] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A Simple and Effective Method to Detect Orthogonal Vanishing Points in Uncalibrated Images of Man-Made Environments. In *Eurographics 2016*, Lisbon, Portugal, May 2016.
- [13] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.
- [14] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

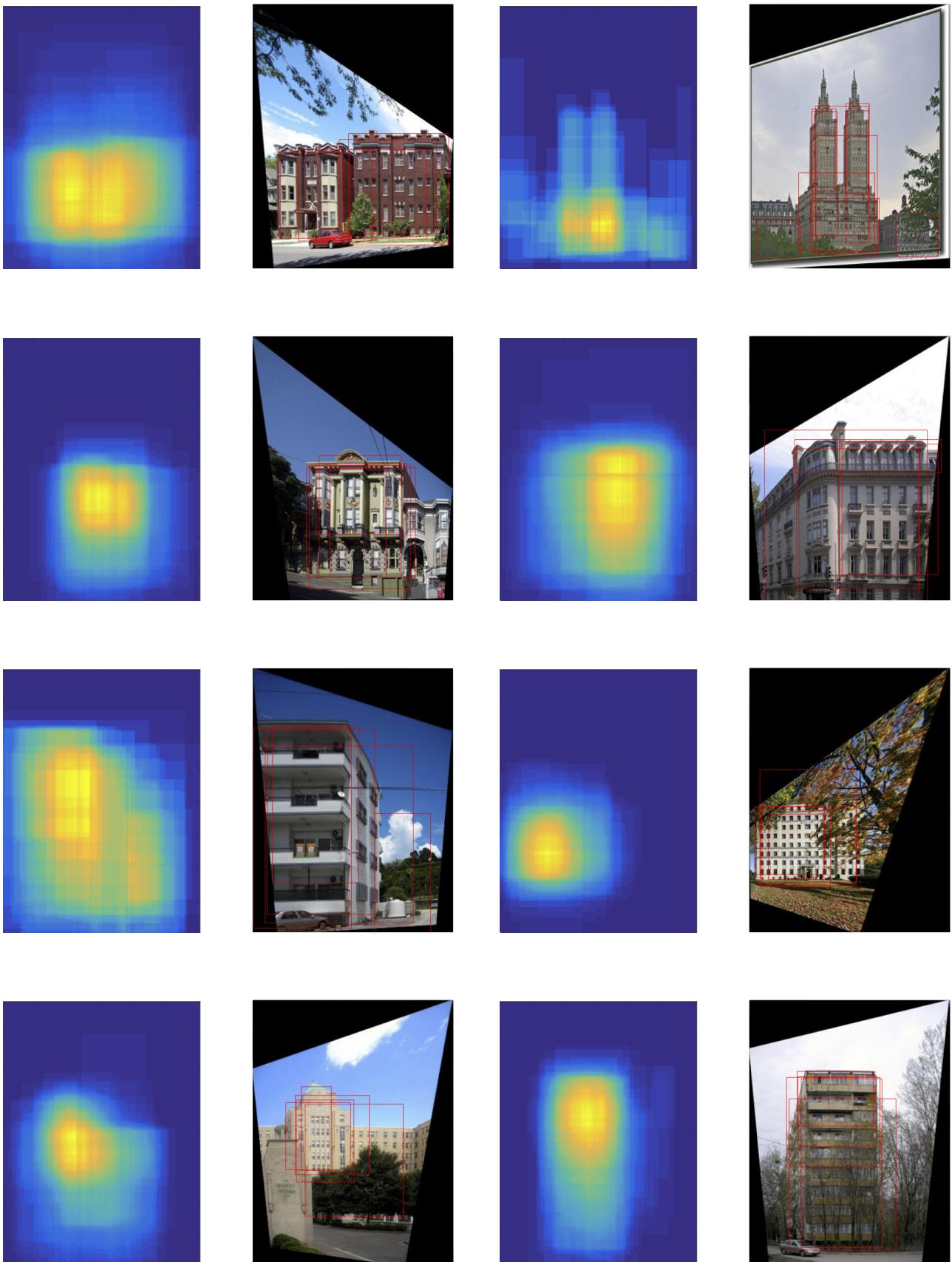


Fig. 12. Map of the overlapped scores after the SVM classification and first ranked windows.