FACADE SEGMENTATION WITH A STRUCTURED RANDOM FOREST

Kujtim Rahmani, Hai Huang, Helmut Mayer

Bundeswehr University Munich, Institute for Applied Computer Science, Visual Computing, Neubiberg, Germany
{kujtim.rahmani, hai.huang, helmut.mayer}@unibw.de

KEY WORDS: Facade, Image interpretation, Structured learning, Random Forest

ABSTRACT:

In this paper we present a bottom-up approach for the semantic segmentation of building facades. Facades have a predefined topology, contain specific objects such as doors and windows and follow architectural rules. Our goal is to create homogeneous segments for facade objects. To this end, we have created a pixelwise labeling method using a Structured Random Forest. According to the evaluation of results for two datasets with the classifier we have achieved the above goal producing a nearly noise-free labeling image and perform on par or even slightly better than the classifier-only stages of state-of-the-art approaches. This is due to the encoding of the local topological structure of the facade objects in the Structured Random Forest. Additionally, we have employed an iterative optimization approach to select the best possible labeling.

1. INTRODUCTION

Facade interpretation is a challenging task in photogrammetry, computer vision and city modeling. Whereas some former work focuses on 3D-interpretation of facades (Mayer and Reznik, 2006, Reznik and Mayer, 2008), the current trend concerning facade modeling from images or image sequences is directed towards pixelwise labeling where in the first step pixels are classified without regard to the structure and the neighborhood of facade objects. As a result, the labeled images are often very noisy (Martinović et al., 2012, Teboul et al., 2010, Cohen et al., 2014, Jampani et al., 2015) and do not follow architectural constraints. Noisy images are often encountered as a result of methods that classify neighboring pixels independently (Fig.1-c). Results that do not follow the architectural constraints are especially produced by methods that classify superpixels (Fig.1-d) with errors, such as sky regions below the roof and balconies at random places on the facade. Thus, it is important that the labeled facade follows the weak architectural constraints. This, e.g., means that the balconies are located under the windows, entrance doors on the first floor and the roof is above the top floor. Opposed to the flexible weak architectural constraints, hard architectural constraints are rules such as that all facade windows have the same dimension and build an even-spaced grid structure or that all balconies have the same dimension.

In this paper, our objective is facade interpretation creating facade objects in the form of homogeneous segments. We achieve this by employing a Structured Random Forest which has the advantage of producing nearly noise free images.

As a baseline classifier this gives on par or even slightly better than current state-of-the-art methods. The challenges are to reduce the noise without compromising the classification performance and to produce a structured representation of the facade objects. We tackle these challenges by taking not only the neighboring pixels, but also the semantic neighborhood into account. To be more flexible, our algorithm does not assume any prior knowledge about the global arrangement of facade objects, such as a grid structure for windows or the same dimensions for all balconies.

We demonstrate that our structured learning algorithm together with problem specific features is efficient for facade segmentation due to its ability to learn the local structure and the dependency of neighboring pixels. The qualitative improvements obtained by the proposed method are shown in Fig. 1 by comparing its results to two related approaches.

Additionally, we employ an iterative optimization approach to improve the accuracy. It chooses the best patch from the candidate patches to label a segment of the facade.

The paper is organized as follows: In Section 2 an overview of related work is given. It is followed by an introduction to Structured Random Forests in Section 3 and its optimization in Section 4. We present our classification algorithm, the employed features and results including an evaluation in Section 5. The paper ends up with a conclusion and recommendation for future work in Section 6.

2. RELEATED WORK

Current approaches for facade image segmentation can be classified into two categories: Top-down methods (Gadde et al., 2016, Teboul et al., 2011), which use shape grammars to segment facades, and bottom-up methods (Cohen et al., 2014, Jampani et al., 2015, Martinović et al., 2012), which employ pixel level classifiers combined with a Conditional Random Field (CRF) or an optimization method.

The top-down methods try to find the best possible facade segmentation using an initial labeling or segmentation and sets of rules which in most cases are hand-crafted. The rules are integrated in a parse-tree and the complete facade is represented as a tree, with the nodes split according to the characteristics of the image and the rules. State-of-the-art methods employing grammars achieve a lower pixelwise accuracy than bottom-up methods. This is due to their large search space and their low efficiency of finding the optimal solution. The only exception of a grammar-based method achieving a high accuracy is (Koziński et al., 2014), but it is very time-inefficient. It needs around 4 minutes for an image of the Ecole Centrale Paris Facades

doi:10.5194/isprs-annals-IV-1-W1-175-2017

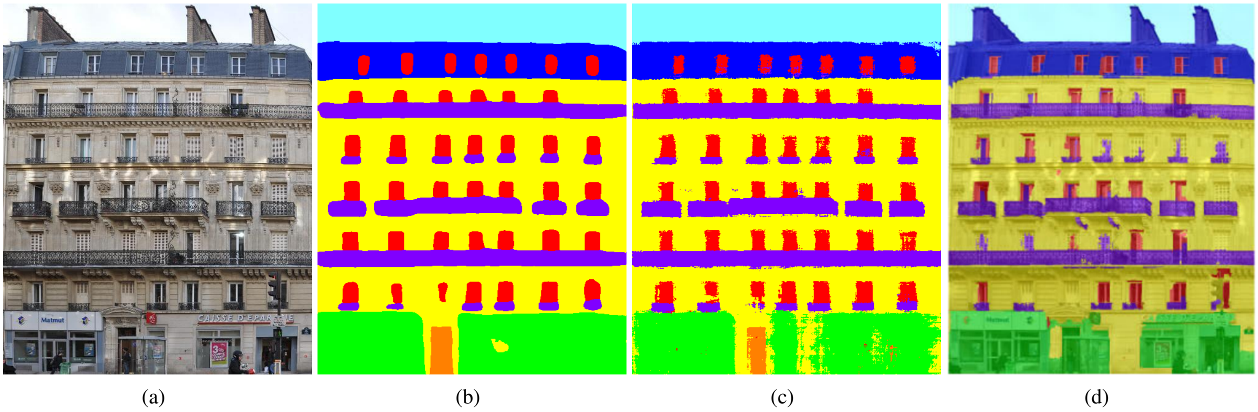


Figure 1. Qualitative results of different methods. (a) shows the input image, (b) the output of our method and (c) the result of (Jampani et al., 2015) which is learned with boosted decision trees after three iterations of Auto-context (Tu, 2008). (d) depicts the output of (Martinović, 2015) which employs superpixel classification. The class labels are described in Fig. 2

Database (ECP) facade dataset (c.f. Section 5.1). In comparison, our approach takes less than 10 seconds per image. (Koziński et al., 2014) first labels the facade using the facade segmentation method presented in (Cohen et al., 2014) and then the facade objects are refined by grammar rules.

Current efficient methods for facade segmentation are based on the bottom-up paradigm. They usually consist of two parts. First, a pixelwise classification algorithm or a superpixel segmentation and classification algorithm with integrated object (window, door and car) detector is trained. In the second step, the labeled output image is optimized, noise reduced and the facade objects are delineated.

In (Cohen et al., 2014) a Dynamic Programming (DP) approach is used in an initial labeled image to find the optimal solution for facade segmentation. They employ hard architectural constraints between facade objects creating an iterative process that step by step defines facade objects. The algorithm is evaluated on the ECP dataset. Balconies and windows are defined in the first DP iteration, doors and shops in the second, and in the last roof and sky. They state that their algorithm performs in most cases very inaccurate for other images. This is because their DP approach encodes a very specific architectural facade style. On the other hand, their DP optimization algorithm is very fast and it takes on average 2.8 seconds per ECP facade image.

(Jampani et al., 2015) presents a classification method using Auto-context (Tu, 2008). They use three iterations of Auto-context in their algorithm and it produces quite accurate results. In their classifier, they employ in each iteration features from the output of the previous iteration. For each pixel they use class distributions, column and row statistics as well as entropy. The classifier consists of boosted decision trees learned by stacked generalization. In the classifier the output of the door and window detectors is added as features. The Auto-context features can encode the density of the predicted class around the pixel as well as global characteristics. They lead to good results, but still the output of the classifier is noisy and contains non homogeneous segments. On average, the system takes around 5 seconds to label an image of the ECP dataset. After three iterations of learning for the decision trees they employ a CRF to optimize and reduce the noise. The last step improves the results for about 1% and it reduces the noise but it takes another 24 seconds. The complete system takes around 30 seconds on average to label an ECP facade image.

In (Martinović et al., 2012) a three layered approach for facade segmentation is described. In the first layer, a Recursive Neural Network (RNN) is trained on an over-segmented facade. The output is a probability distribution for each pixel for all classes. In the second layer, window and door detectors are introduced. The output of the the detectors and the RNNs are incorporated in a Markov Random Field (MRF). The result of the MRF is still a noisy labeled image and does not follow the weak architectural constraints. Thus, in the top layer they introduce weak and hard architectural constraints. The employed energy minimization method uses the architectural constraints as well as very particular architectural characteristics of the dataset such as that the second and the fifth floor have a running balcony.

The above authors have extended their work to the ATLAS approach (Mathias et al., 2016). In the first layer the facade is segmented into superpixels, which are labeled by a classifier with one of the facade classes. Different segmentation methods and classifiers such as Multiclass Logistic Regression, CRF, Multilayer Perceptron, Multiclass Support Vector Machine (SVM) and RNN have been evaluated. Their best first layer, using SVM, achieves an accuracy of 84.75% on the ECP dataset which is an improvement of around 2% compared to their previous work. The superpixel classification gives significantly higher accuracy for shops, roof and sky, but reduces the accuracy for doors. Thus, the SVM segmentation classification method is better in defining large objects and worse for small and less homogeneous objects. In the second layer, they improve their object detectors and introduce a car detector. The last layer is similar to (Martinović et al., 2012). Overall, the final system has a pixelwise accuracy of 88.02%, which is an improvement of 3.85% compared to their previous work. The three layers take around 180 seconds to label an image of the ECP dataset.

3. STRUCTURED RANDOM FOREST

In this section we give an overview of Decision Trees and the Structured Random Forest (Kontschieder et al., 2011a, Kontschieder et al., 2014). The Structured Random Forest is employed to classify each pixel by using the region around it considering its local structure. The particular strength of this classifier is that it outputs a nearly noise-free labeled image.

3.1 Decision Tree

The Decision Tree and Random Decision Forest are introduced and notations are given based on (Kontschieder et al., 2011a) and (Dollár and Zitnick, 2013).

A Decision Tree can be represented as $f_t(x) = y$ where x is an n-dimensional sample classified recursively by branching down to the leaf node depending on the feature values of the sample x. The leaf node assigns the class $y \in Y$ to the data sample x. For our problem the data samples x are image patches and their channel features, and the class labels are patches with pixels labeled with facade objects $Y = "window", "door", "wall", "sky", etc.$

Each node of the tree decides if it branches left or right based on the split function defined as

$$h(x, \theta_j) \in \{0, 1\} \tag{1}$$

with parameters θ_j. If the function $h(x, \theta_j)$ returns 0, then the node j is traversed down to the left, otherwise to the right. The recursive traversal terminates at the leaf nodes. The output of the tree for the sample x is the prediction stored in the leaf node where the traversal process terminated and a target label $y \in Y$ or a distribution over the labels of Y.

The split function (Equation 1) can be complex. Yet, the most frequent choice is a simple function, where a single feature dimension of sample x is compared to a threshold. Formally $\theta = (k, \tau)$ where k defines the feature of the sample and τ the threshold. Then, $h(x, \theta) = [x(k) < \tau]$, with $[\cdot]$ the indicator function. Another often used split function is $h(x, \theta) = [x(k_1) - x(k_2) < \tau]$ with $\theta = (k_1, k_2, \tau)$.

3.2 Random Decision Forest

A Random Decision Forest is an ensemble of T independent trees f_t. For a sample x of the dataset D the Random Decision Forest F of T trees gives the prediction as a combination of the $f_t(x)$ with $t \in 1, 2, ..T$ using an ensemble model such as majority vote or consensus. In the leaf nodes of each tree of F arbitrary information about the model, which can help in the final decision of the prediction, is stored. The leaf node reached in each tree of F depends only on the feature values of x. The prediction of the Random Forest can be generated in an efficient way. The above characteristics allow to model complex output spaces, such as the structured outputs in (Kontschieder et al., 2011a)

3.3 Training of Decision Trees

The main goal of the training of trees is to find the best split functions (Equation 1) so that an as high as possible accuracy for the whole classification system is achieved. Formally, for a tree node j and training set $S_j \subseteq X \times Y$ we define parameters θ_j of the split function (Equation 1) to maximize the "quality" of the split. One possible measure of the split quality is the information gain

$$I_j = I(S_j, S_j^L, S_j^R) , \tag{2}$$

where $S_j^L = (x, y) \in S_j \mid h(x, \theta_j) = 0$, $S_j^R = S_j \setminus S_j^L$. The split parameters θ_j are chosen to maximize I_j. The training is recursively conducted for the left node with the data S_j^L and the right node with S_j^R. It ends when the predefined depth of the tree has been reached or the size of S_j falls below a predefined threshold. For a classification with multiple classes, the information gain is defined as:

$$I_j = H(S_j) - \sum_{k \in \{L,R\}} \frac{\mid S_j^k \mid}{\mid S_j \mid} H(S_j^k) \tag{3}$$

where $H(S_j) = \sum_y log(p_y)$ denotes the Shannon entropy over the probability distribution p_y of the class elements in the subset S_j.

3.4 Structured Random Forest

During the pixel labeling of the facade images, it is important to consider the local and global arrangement of the facade objects. There are architectural constraints and an object hierarchy that can be encoded in the classification algorithm.

In Standard Random Forests each pixel is assigned to a single class label independently of the neighboring pixels. This is the main cause that Standard Random Forests produce a noisy labeling. The adaptation of structured learning (Nowozin and Lampert, 2011) to random forests allows to output a labeled patch and, thus, to strongly reduce the noise in image segmentation or labeling.

In comparison with Standard Random Forests, the Structured Random Forests output for each leaf node a patch with predefined dimensions and during the traversal of the tree the class labels of the neighboring pixels are also considered. The first means that each pixel is labeled from multiple patches, i.e., the label information is shared with the neighbors and the latter means that the split functions are constructed from two or more pixels.

Structured Random Forests struggle with the exponentially increased complexity of the output space. To overcome this drawback, various methods have been proposed for the test and training function selection, such as Principal Component Analysis (PCA) and probabilistic approaches (Kontschieder et al., 2014). In our approach the output of a leaf node is computed as a joint probability distribution of the labels assigned to the leaf node.

We use a training selection method similar to (Kontschieder et al., 2011a). It selects the best split function at each node based on the information gain with respect to two label joint distributions. The training works as follows: Let S_t be the subset of the patches with dimension $2d + 1 \times 2d + 1$ and center $(0, 0)$ that is to be split at a tree node (t). We select the center pixel and a pixel in a random position (i, j) where $\mid i \mid \leq d$ and $\mid j \mid \leq d$ of the patch and compute the best split for S_t for a feature $x(k)$. The best split is chosen by computing the information gain of S_j. This continues recursively until the leaf node is reached (Kontschieder et al., 2011a, Kontschieder et al., 2014, Nowozin and Lampert, 2011).

4. OPTIMIZATION FOR STRUCTURED RANDOM FORESTS: THE LABEL PUZZLE GAME

Since the Structured Random Forest outputs a patch with predefined dimensions $d \times d$, each pixel is covered by d^2 patches. Thus, each pixel is assigned d^2 values distributed over the classes. The pixel label is selected by majority vote.

Let l be the labeled image I from the majority vote. We employ an iterative optimization method (Kontschieder et al., 2011b) to select the best possible labeling for an image from the output patches of the trees of the forest. Let the Forest F have T trees and let the patch $p(t)_{(i,j)}$ be the predicted patch of the tree t for the pixel location with patch center (i, j). We define *agreement* of the patch $p(t)_{(i,j)}$ as the number of correctly predicted pixels in the image l labeled by the previous step.

$$\phi^{(i,j)}(p(t), l) = \sum_{(r,c)\in p(t)} [p(t)_{(i,j)}(r, c) = l(r, c)] \quad (4)$$

For the next iteration, to label the patch with the center at the pixel $I(i, j)$ the tree with the highest agreement score $\phi^{(i,j)}(p(t), l)$ is selected. In other words, from the forest F for each pixel the tree is chosen that has the most accurately labeled pixel in the predifined neighborhod $(d \times d)$. We perform this operation iteratively. For this iterative model it has been proven that it converges to a local maximum with regard to the agreement function. For the complete proof and description of the method we refer to (Kontschieder et al., 2011b).

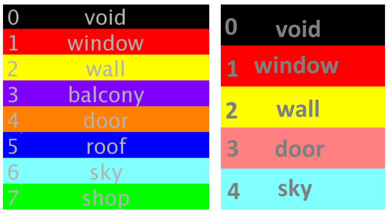
5. EXPERIMENTS

5.1 Datasets

ECP This dataset contains 104 rectified facade images of Hausmannian architectural buildings from Paris. The images are labeled according to seven semantic classes (*balcony, door, roof, shop, sky, wall and window* – Fig. 2). The dataset has two versions: The first version is presented by (Teboul et al., 2011) and the second by (Martinović et al., 2012). In the first version all windows have the same width and are aligned on a grid. The second dataset is more accurate and provides the real position of the objects. In both datasets doors, windows, and shops are labeled as rectangles.

ParisArtDeco This dataset is used and published by (Gadde et al., 2016). It contains 79 rectified facade images from Paris. As for the ECP dataset, the images are labeled according to seven semantic classes. We use this dataset only to train the door and window detectors.

Graz It contains 50 rectified facade images of different architectural styles (Classicism, Biedermeier, Historicism, and Art Nouveau) from Graz (Riemenschneider et al., 2012). The images are labeled according to four semantic classes (*door, sky, wall and window* – Fig.2).



(a) (b)

Figure 2. Label sets and colors for (a) ECP dataset (b) Graz dataset

5.2 Image Features

As image features we use RGB color, CIELab raw channels, Histogram of Oriented Gradients (HOG), location information, 17 TextonBoost filter responses and Local Binary Patterns. We additionally employ the per pixel score of the door and window detector of (Jampani et al., 2015), which is trained based on the integral channel feature detector (Dollar et al., 2009). The output are bounding boxes with corresponding scores and the scores are assigned to each pixel in the bounding box. In contrast to (Jampani et al., 2015) which trains the door and window detectors on the same dataset, we train them on the ParisArtDeco dataset and use it for the ECP and Graz dataset.

Since the facades are rectified, we add the per row variance and the median of the RGB channels as well as for each pixel the corresponding deviation from the median value. We also employ the correlation coefficients between covariances of RGB raw channel intensities.

5.3 Evaluation

Our algorithm has been evaluated on the ECP and the Graz dataset. For both datasets we show the five-fold cross-validation results. The ECP dataset is divided in four sets with 20 images and one set with 24 images and the Graz dataset is divided in five sets of 10 images. To compare the results with (Jampani et al., 2015), we use the same configuration of sets. The Structured Random Forest is trained with a patch size of 15 × 15 pixels and a termination threshold of five samples for the leaf node.

Our empirical results (Fig. 4) are compared for the ECP dataset with the current four best bottom-up methods and for the Graz dataset (Fig. 3) with the results obtained by the method presented in (Jampani et al., 2015). We compare the two phases classification and optimization of our algorithm separately with the current state-of-the-art approaches.

Tables 1 and 3 give the evaluation results for the classifiers without optimization. We observe that our classification algorithm performs on par or even slightly better than the current best classifiers. Particularly, our algorithm is significantly better for doors. Overall our classifier is better by about 1% than the other algorithms and produces results with less noise.

The optimization method relies just on the basic local arrangement of the facade objects. The global arrangement is not considered. Because of the local optimization, our algorithm delineates the objects efficiently (Fig. 5). Since we do not make any assumption about the global features and global arrangement of the facade objects, our algorithm is quantitatively (Tables 2 and 4) very slightly worse than the current state-of-the-art approaches.

Class	Results	
	I	ours
Door	57.3	**75.71**
Window	78.2	**78.63**
Wall	94.9	**95.39**
Sky	**87.4**	73.85
Average	79.47	**80.9**
Overall	90.18	**90.34**

Table 1. Labeling results of the classification stage on dataset Graz. Column (I) gives results from the first stage of Auto-context (Jampani et al., 2015). Ours: Structured Random Forest only. Best results are given in bold.

6. CONCLUSION AND FUTURE WORK

We have presented a method for facade image segmentation resulting into semantic regions and their delineation. We employ

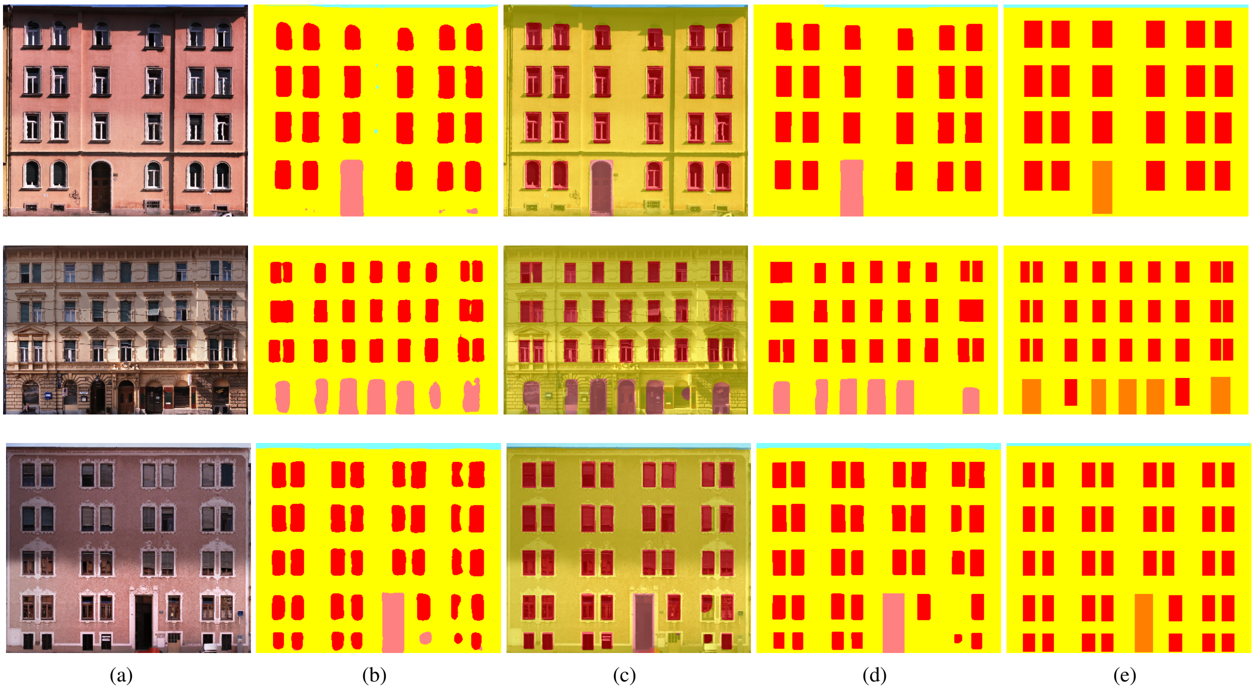


Figure 3. Qualitative results on the Graz dataset. The facade segments are homogeneous and nearly without noisy. Column (a) input images, (b) results from Structured Random Forest, (c) results after 10 iterations of the optimization algorithm mapped into the input image, (d) results after 50 iterations and (e) ground truth. Colors see Fig. 2.

Class	Results			
	I-1	I-2	ours-10	ours-50
Door	62.7	63	78.31	**79.08**
Window	**81.5**	80.9	79.48	79.27
Wall	94.9	95.8	95.81	**96.03**
Sky	90.5	**90.6**	74.57	75.76
Average	82.42	**82.56**	82.05	82.54
Overall	91.16	**91.68**	90.93	91.13

Table 2. Labeling results after optimization on dataset Graz. Column (I-1) labeling optimized with three iterations of Auto-context (Jampani et al., 2015). (I-2) labeling optimized with Auto-context and CRF with an 8-connected neighborhood and pairwise Potts potentials. Ours-10 and Ours-50 after 10 and 50 iterations, respectively. Best results are given in bold.

a Structured Random Forest for facade labeling and have shown that the structured learning is efficient, because of its ability to learn the local structure of the facade objects. The experimental results on the ECP and Graz facade datasets demonstrate that it performs on par concerning certain aspects or even slightly better than state-of-the-art approaches. We presented an optimization algorithm which iteratively converges to the best possible labeling. In the future we want to extend the optimization algorithm by adding constraints on the global structure of the facade and we want to speed it up. Additionally, we plan to develop a better object detector and features more specific for the problem.

7. ACKNOWLEDGEMENTS

This paper is supported by grant MA 1651/15 from Deutsche Forschungsgemeinschaft (DFG).

Class	Results					
	I	II-1	II-2	III-1	III-2	Ours
Door	76.2	43	60	41	58	**83.77**
Shop	87.6	79	86	91	**97**	94.61
Balcony	**85.8**	74	71	75	81	85.68
Window	77.0	62	69	64	76	**79.20**
Wall	91.9	91	**93**	91	90	91.32
Sky	**97.3**	91	91	94	94	95.98
Roof	86.5	70	73	82	87	**90.76**
Average	86.04	72.85	77.57	86.67	83.36	**88.76**
Overall	88.86	82.63	85.06	84.75	88.07	**89.87**

Table 3. Labeling results of the classification stages of various approaches on dataset ECP. Column (I) gives results for the first stage of Auto-context (Jampani et al., 2015). (II-1) and (II-2) present results for the first and the second layer of the three-layered approach (Martinović et al., 2012), respectively. (III-1) and (IV-1) show results for the first and the second layer of ATLAS (Mathias et al., 2016). Best results are marked in bold.

REFERENCES

Cohen, A., Schwing, A. G. and Pollefeys, M., 2014. Efficient structured parsing of facades using dynamic programming. In: *Computer Vision and Pattern Recognition*, pp. 3206–3213.

Dollár, P. and Zitnick, C. L., 2013. Structured forests for fast edge detection. In: *International Conference on Computer Vision*, pp. 1841–1848.

Dollar, P., Tu, Z., Perona, P. and Belongie, S., 2009. Integral channel features. In: *British Machine Vision Conference*, pp. 1–11.

Gadde, R., Marlet, R. and Paragios, N., 2016. Learning grammars for architecture-specific facade parsing. *International Journal of Computer Vision* 117(3), pp. 290–316.

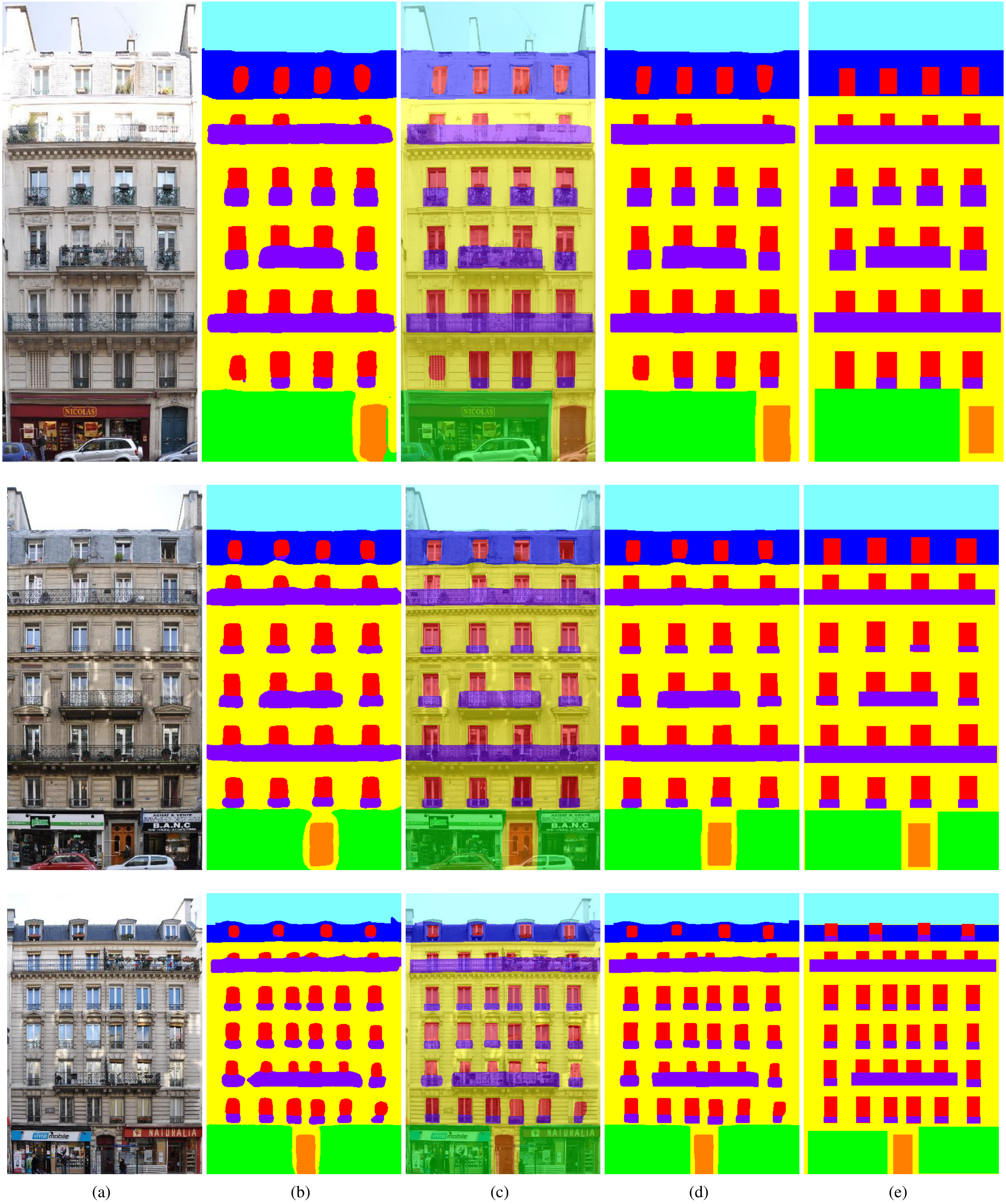
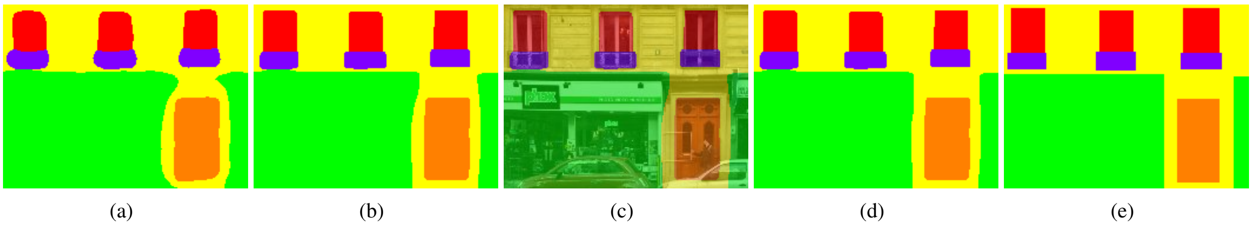


Figure 4. Qualitative results on the ECP dataset. The facade segments are homogeneous and nearly without noise. Column (a) input images, (b) results from Structured Random Forests, (c) results after 10 iteration of the optimization algorithm mapped in the input images, (d) results after 50 iterations of the algorithm, (e) ground truth. Colors see Fig. 2.

Class	Results						
	I-1	I-2	II	III	IV	Ours-10	Ours-50
Door	80.4	81.3	67	71	79	**82.16**	79.47
Shop	91.6	93.2	93	95	94	94.85	**95.17**
Balcony	89.4	89.3	70	87	**91**	88.38	86.43
Window	81.8	82.3	75	78	**85**	80.54	80.41
Wall	92.4	**92.9**	88	89	90	91.47	91.52
Sky	98.0	**98.2**	97	96	97	96.15	96.18
Roof	88.1	89.2	74	79	**93**	90.98	91.02
Average	88.79	**89.49**	80.71	85.22	89.4	88.93	88.60
Overall	90.81	**91.42**	84.17	88.02	90.82	90.21	90.24

Table 4. Labeling results after optimization for various approaches on dataset ECP. Column (I-1) gives results for Auto-context (Jampani et al., 2015) after three iterations and (I-2) for Auto-context with CRF with an 8-connected neighborhood and pairwise Potts potentials. (II) presents the third layer of the three layered approach (Martinović et al., 2012) for facade parsing and (III) ATLAS (Mathias et al., 2016), respectively. (IV) gives results for the DP optimization approach (Cohen et al., 2014). The end results of our method after 10 and 50 iterations are given by Ours-10 and Ours-50, respectively. Best results are marked in bold.



(a) (b) (c) (d) (e)

Figure 5. Iterative delineation. Image (a) results from the Structured Random Forest. (b) gives results after 10 iterations, (c) results after 10 iterations mapped into the input image, (d) results after 50 iterations and (e) ground truth. One can observe that the object delineation is very accurate even after only the 10th iterations. Colors see Fig. 2.

Jampani, V., Gadde, R. and Gehler, P. V., 2015. Efficient facade segmentation using auto-context. In: *IEEE Winter Conference on Applications of Computer Vision*, pp. 1038–1045.

Kontschieder, P., Bulo, S. R., Bischof, H. and Pelillo, M., 2011a. Structured class-labels in random forests for semantic image labelling. In: *International Conference on Computer Vision*, pp. 2190–2197.

Kontschieder, P., Bulò, S. R., Donoser, M., Pelillo, M. and Bischof, H., 2011b. Semantic image labelling as a label puzzle game. In: *British Machine Vision Conference*, pp. 1–12.

Kontschieder, P., Bulo, S. R., Pelillo, M. and Bischof, H., 2014. Structured labels in random forests for semantic labelling and object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(10), pp. 2104–2116.

Koziński, M., Obozinski, G. and Marlet, R., 2014. Beyond procedural facade parsing: Bidirectional alignment via linear programming. In: *Asian Conference on Computer Vision*, pp. 79–94.

Martinović, A., 2015. Invers procedural modeling. Technical report, Arenberg Doctoral School Faculty of Engineering Science, KU Luven.

Martinović, A., Mathias, M., Weissenberg, J. and Van Gool, L., 2012. A three-layered approach to facade parsing. In: *European Conference on Computer Vision*, pp. 416–429.

Mathias, M., Martinović, A. and Van Gool, L., 2016. ATLAS: A Three-Layered Approach to Facade Parsing. *International Journal of Computer Vision* 118(1), pp. 22–48.

Mayer, H. and Reznik, S., 2006. MCMC Linked with Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. (36) 3, pp. 130–135.

Nowozin, S. and Lampert, C. H., 2011. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision* 6(3–4), pp. 185–365.

Reznik, S. and Mayer, H., 2008. Implicit Shape Models, Self Diagnosis, and Model Selection for 3D Facade Interpretation. *Photogrammetrie – Fernerkundung – Geoinformation* 3/08, pp. 187–196.

Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D. and Bischof, H., 2012. Irregular lattices for complex shape grammar facade parsing. In: *Computer Vision and Pattern Recognition*, pp. 1640–1647.

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P. and Paragios, N., 2011. Shape grammar parsing via reinforcement learning. In: *Computer Vision and Pattern Recognition*, pp. 2273–2280.

Teboul, O., Simon, L., Koutsourakis, P. and Paragios, N., 2010. Segmentation of building facades using procedural shape priors. In: *Computer Vision and Pattern Recognition*, pp. 3105–3112.

Tu, Z., 2008. Auto-context and its application to high-level vision tasks. In: *Computer Vision and Pattern Recognition*, pp. 1–8.