HIGH QUALITY FACADE SEGMENTATION BASED ON STRUCTURED RANDOM FOREST, REGION PROPOSAL NETWORK AND RECTANGULAR FITTING

Kujtim Rahmani, Helmut Mayer

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

Commission II, ICWG II/III

KEY WORDS: Facade Segmentation, Model Fitting, CNN, Object Detection

ABSTRACT:

In this paper we present a pipeline for high quality semantic segmentation of building facades using Structured Random Forest (SRF), Region Proposal Network (RPN) based on a Convolutional Neural Network (CNN) as well as rectangular fitting optimization. Our main contribution is that we employ features created by the RPN as channels in the SRF. We empirically show that this is very effective especially for doors and windows. Our pipeline is evaluated on two datasets where we outperform current state-of-the-art methods. Additionally, we quantify the contribution of the RPN and the rectangular fitting optimization on the accuracy of the result.

1. INTRODUCTION

Facade segmentation is an important part of urban scene understanding and 3D building reconstruction and of interest for architectural design, movies and video games.

Early work on facade segmentation has focused on window detection and 3D reconstruction from multiple images (Mayer and Reznik, 2006, Reznik and Mayer, 2008). In recent years, most work on facade segmentation is based on single images (Cohen et al., 2014, Jampani et al., 2015, Mathias et al., 2016, Rahmani et al., 2017, Schmitz and Mayer, 2016). Additionally, the Varcity dataset (Riemenschneider et al., 2014) has been published focusing on facade image and facade point cloud labeling. As the above papers, we also concentrate on single facade image segmentation.

The biggest challenges of the basic pixel-wise segmentation are noise and the complex shapes of facade objects such as doors, windows and balconies. The first problem particularly concerns algorithms that classify each pixel or superpixel independently of its neighbors. It can be reduced by incorporating interaction with the neighborhood. The latter problem is dealt with by model fitting and by making use of the global structure of facade objects and architectural constraints. Some approaches encode hard architectural constraints in their algorithms, such as a grid window structure and that all balconies have the same dimension. Others employ soft architectural constraints like that the roof is on top of the building, or that shops are on the first floor. An additional challenge of facade segmentation is the variety of building types and architectural styles, which leads to different shapes and arrangements of the facade objects.

The main contribution of this paperis that we introduce a Region Proposal Network (RPN) to create proposals for objects such as window, door, balcony, shop and sky together with their corresponding probability. These probabilities are transformed into features which are input to Structured Random Forest (SRF) (Kontschieder et al., 2014) classification. This leads to a segmentation with very few noise. Finally, a deterministic rectangular fitting is used to create rectangularly shaped facade objects and a grid structure.

The pipeline (Fig. 1) presented in this paper outperforms all other state-of-the-art approaches on the current benchmarks without relying on hard architectural constraints. To clarify the importance of the introduction of the RPN we introduce the RPN to facade segmentation and quantify its contribution to the good overall performance. The high quality results of RPN and SRF are supplemented by a fast and yet accurate model fitting.

The paper is organized as follows: In the next section we give an overview of related work. Sections 3, 4 and 5 describe in depth our pipeline SRF, RPN and rectangular fitting, respectively. Experiments and the technical details are given in Section 6. Finally, we present the evaluation, draw conclusions and point to future work.

2. RELATED WORK

We distinguish two types of facade segmentation methods: Grammar based (top-down) and classification-based (bottom-up) methods. Top-down methods usually first classify each pixel or generate facade object hypotheses. Then they use shape grammars to parse the facade images. They learn the hierarchy and distribution of facade objects as well as the architectural characteristics of the data set. Because of this they can predict object positions, particularly for windows, even when they are occluded by vegetation or other objects

From a processing perspective, top-down methods first divide the facade images in bigger parts and then recursively split each part in smaller facade objects. The division rules are hand crafted or learned and integrated into a shape parse tree grammar.

State-of-the-art grammar based methods usually achieve an accuracy of pixel-wise classification below 85% (Gadde et al., 2016) on the ECP benchmark dataset (Teboul et al., 2011). A problem of grammar based methods is their time inefficiency during training and inference, where they need several minutes to process a typical image (Koziński et al., 2015, Gadde et al., 2016).

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-223-2018 | © Authors 2018. CC BY 4.0 License.

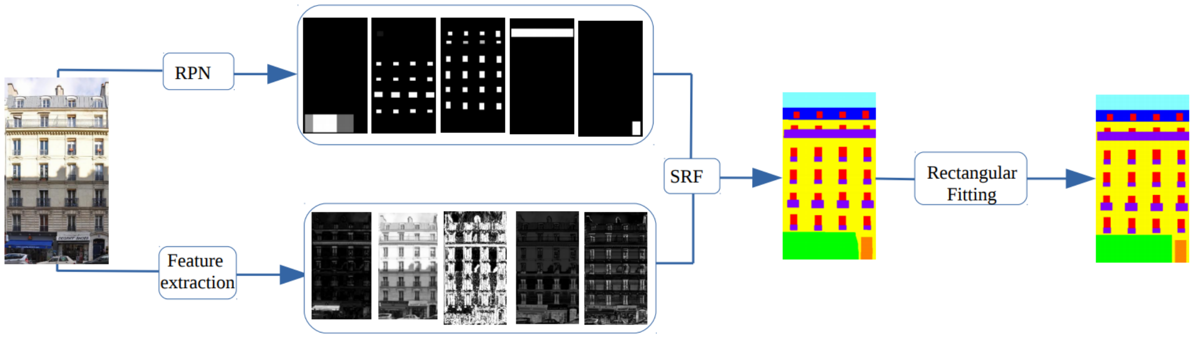


Figure 1. Architecture of the proposed pipeline for facade parsing (RPN – Regional Proposal Network, SRF – Structured Random Forest)

Currently, the most efficient but also highest quality methods are bottom-up. They employ a pipeline architecture which contains pixel (or superpixel) labeling, object detectors and optimization. Each part of the pipeline tries to correct wrongly classified pixels or to optimize the segments created by previous parts.

Dynamic programming (DP) is applied in (Cohen et al., 2014) to segment facade objects. Pixel-wise classification is used and hard architectural constraints are encoded as constraints in the DP. At each of multiple steps of DP optimization a combination of facade objects is used. In the DP, e.g., the constraints that windows and balconies are rectangular and that balconies on the same row usually have the same dimensions are employed. In the following steps the shop, door, roof and sky segments are optimized. In (Cohen et al., 2017) the same authors additionally make use of the symmetry of the facade. This reduces problems with occlusions and improves the accuracy.

Auto-context (Tu, 2008) is used in (Jampani et al., 2015). The pipeline consists of boosted random trees, object detectors and conditional random fields (CRF). First, an object detector for doors and windows is trained and the scores of the detectors are used as features in the boosted random trees. Additionally, for each pixel 761 low level features such as TextonBoost filters, Histogram of Oriented Gradients (HOG), Local Binary Pattern features and averages over image rows and columns are computed.

For building an object detector, the Integral Channel Features detector (Dollar et al., 2009) is employed, which outputs bounding boxes. The score of each pixel is the sum of the scores of each bounding box that contains the pixel. Additionally, three iterations of the Auto-context algorithm are used. In each iteration the output of the previous step is added as well as features such as the class probability for each pixel, entropy, row and column statistics, distance to the nearest class pixel, color model per class, bounding box features and neighborhood statistics. The iterations improve the accuracy by 1% to 8% on the benchmarked dataset (Jampani et al., 2015). As postprocessing, a CRF is employed to delineate and reduce the noise of the segments. This improves the accuracy by no more than another 1% but adds more than 24 seconds. Compared with less than 6 seconds for all previous steps, the CRF takes more than 80% of the time.

(Martinović et al., 2012) presents a three layered approach. On the first layer the image is segmented with a Recursive Neural Network which outputs the class probability distribution of each pixel. The second layer consists of a door and window detector and a Markov Random Field (MRF). Since the results are still noisy and only moderately accurate, a third layer is introduced. It consists of an energy minimization incorporating architectural constraints as well as characteristics of the datasets such as that the second and the fifth floor must have a running (long) balcony.

The above work is extended in the ATLAS approach (Mathias et al., 2016). First, the image is segmented into superpixels. A range of segmentation methods such as RNN, Perceptron, Multiclass Support Vector Machine (SVM), Multiclass Logistic Regression and CRF have been tried and it has been shown that the SVM works best. The results show that the first layer leads to an improvement of 2% compared to the previous approach. This improvement is due to a significantly higher accuracy for the bigger classes like shop and roof. On the other hand, the accuracy is lower for doors. In the second layer, the door and window detectors are improved and a detector for cars which often occlude the lower part of facades is added. The final postprocessing layer is similar to (Martinović et al., 2012).

(Rahmani et al., 2017) introduces an SRF demonstrating the good performance for facade segmentation. The method outperforms the current classification methods, but the employed iterative optimization algorithms does not perform well quantitatively compared to other optimization approaches.

In the following we describe our novel approach which can be considered as an extension and improvement of (Rahmani et al., 2017).

3. BASICS OF STRUCTURED RANDOM FORESTS

In this section we present a short introduction of Decision Trees and Structured Random Forests (Kontschieder et al., 2014) as well as their advantages for facade segmentation. For more details, please refer to (Kontschieder et al., 2011a, Kontschieder et al., 2011b, Kontschieder et al., 2014). The main difference between traditional Random Forests and Structured Random Forests is that the output of the SRF is an image patch, while the traditional output is just a single pixel.

3.1 Decision Trees

A Decision Tree (DT) is a classification algorithm that accepts as input an n-dimensional feature vector x from a dataset D and outputs a class label $y \in Y$, where Y is the set of class labels. Formally, we can represent a DT by $f_t(x) = y$. A DT classifies a sample recursively by branching down to a leaf node. At each node of a DT a split function is learned deciding how to traverse down until the leaf node is reached. The leaf node assigns a class label to the sample. In facade segmentation the data samples are the pixel's features and the set of labels is: $Y = \{"window", "door", "wall", sky", ..\}$

Each node decides based on the learned split function which is defined as follows (with parameters θ_j)

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

The sample will continue its path to the left node if the output of the split function is 0, otherwise it continues to the right.

3.2 Training of Decision Trees

During training DTs try to find the best split function (Equation 1) so that they achieve the highest possible accuracy. Formally, the set $S_j \subseteq X \times Y$, which has “arrived” at tree node j should be split in two subsets by the split function in a way that the quality of the split is maximized. Often the measure of 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 DT selects the parameters θ_j that maximize the information gain.

The parameters of the split function are usually chosen randomly for a certain number (often up to three) of features and their corresponding thresholds. This process is repeated several times and the combination of features and their corresponding thresholds that maximize the information gain I_j (Equation 2) are chosen as parameters of the split function.

3.3 Random Forests

A Random Forest is a set of T independent DTs. To classify a sample, Random Forests accumulate the T predictions of each tree. From these labels the Random Forests choose a single label, usually by majority vote or consensus. DTs have the problem of overfitting which the redundancy of several trees helps to reduce.

3.4 Structured Random Forests

For facade segmentation it is beneficial to consider the local and global structures of the facade objects. The features and the segmentation algorithm which embody the architectural constraints as well as the object hierarchy. For this, we use SRFs, as they encode the local structure of the objects in their split nodes.

Standard Random Forests label each pixel independently of its neighborhood, leading to labeling to labels with a lot of salt-and-pepper noise. The adaptation of structured learning (Nowozin and Lampert, 2011) to random forests produces as output a patch. This results into almost noise free segments and highly accurately labeled images.

Additionally, SRF have the advantages, that they output for each pixel a patch and that during training also the labels of the neighboring pixels are used. For each pixel multiple labels are proposed from the neighboring pixels and during training the neighborhood is integrated in the split function.

Unfortunately, the patches lead to an exponential increase of the output space compared to the Standard Random Forests. To overcome this problem, different reduction techniques are employed for the output space such as Principal Component Analysis (PCA) (Dollár and Zitnick, 2013) and probabilistic approaches such as (Kontschieder et al., 2014). In this paper a representative patch is computed for each node as the joint probability distribution of the labels assigned to a leaf node.

We use a training methodology similar to (Kontschieder et al., 2011b). The best split parameters are chosen based on the information gain of up to three joint distributions. The training procedure works as follows: Let S_t be the subset of sample patches that has reached the node t. Each sample of S_t has dimension $d \times d$ with center $(0, 0)$. We randomly choose up to three positions (i, j) around the center patch with $|i| \leq n$ and $|j| \leq n$ (n is a chosen neighborhood) as well as a feature and a threshold for each position. The information gain is evaluated for 400 to 1000 randomly chosen combinations of up to three positions, features and thresholds and the best parameters for S_t are chosen. This is repeated recursively until the leaf node is reached.

3.5 Optimization for Structured Random Forests

Each patch is of dimension $d \times d$ and we evaluate every pixel, meaning that each pixel is covered by d^2 patches (except pixels at the borders of the image). The d^2 values are distributed over the classes and the final pixel label is chosen by majority vote.

We use an iterative optimization method (Kontschieder et al., 2011b) which produces sharper edges for the segments, a higher accuracy and removes noisy small segments. It selects the best labeling from the set of patches for each tree of the forest.

Formally, let a training image I with labels l be given, let the SRF F be defined as a set of T structured decision trees and let the tree $t \in T$ for pixel at position (i,j) predict the patch $p(t)_{(i,j)}$. We define an optimization function *agreement score* counting the number of correctly predicted pixels of the patch $p(t)$ on the labeled image l.

$$\phi^{(i,j)}(p(t), l) = \sum_{(r,c) \in p(t)} [p(t)_{(i,j)}(r, c) = l(r, c)] \tag{3}$$

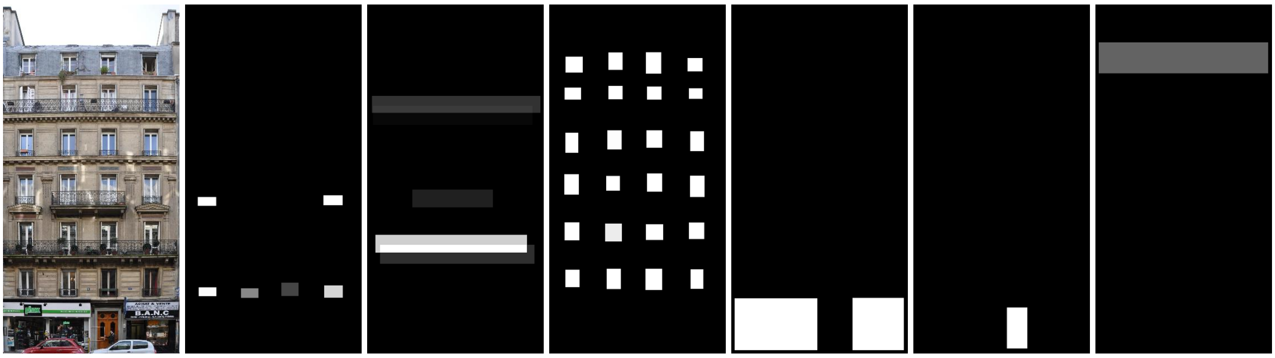
When labeling the patch with center at position $I(i, j)$, the patch from the tree that has produced the highest agreement score is selected as representative patch. Other trees with lower agreement scores are ignored. This step is performed for every pixel with d^2 proposals and the class label for each pixel is chosen through majority vote. The optimization can be performed multiple times until convergence. For the complete proof and more details we refer to (Kontschieder et al., 2011b)

This iterative technique tries to shape the object boundaries similar to the objects that the SRF has “seen” during training.

4. REGION PROPOSAL NETWORK

With the recent advances in Convolutional Neural Networks (CNN) (LeCun et al., 1990), the accuracy of object detection and Region Proposal Network (RPN) algorithms has considerably improved (Ren et al., 2015, Liu et al., 2016). Large data sets such as ImageNet (Krizhevsky et al., 2012) and COCO (Lin et al., 2014) have been made available for training and testing.

We use a pretrained model as basis to train our RPN. We employ the Single Shot Detector-300 (SSD-300) (Liu et al., 2016) as RPN for classes window, door, balcony, long (running) balcony, shop, roof and sky. For each object a separate feature (channel) is created (Figs. 1 and 2). The detectors output rectangles with an attached probability for the existence of the object. Each pixel in the box is given the probability value of the object mapped to the [0,255] range with “inverse min-max normalization”. In the ECP dataset, the RPN produces six features (Fig. 2). During



(a) (b) (c) (d) (e) (f) (g)

Figure 2. Region Proposal Network (RPN). The output of the RPN for the facade objects in image (a) for (b) balconies, (c) long balconies, (d) windows, (e) shops, (f) doors and (g) roof. The intensity represents the confidence of the network in the existence of an object at that position. The brighter the region, the higher is the probability of the existence of the object.

the experiments we realized that with a small addition of space around an object (padding) during the training phase, the object detector performs better. This is particularly helpful for entrance doors which are usually surrounded by the wall making it easier for the RPN to identify them. Additionally, this helps to distinguish entrance doors from shop doors, because the latter are mainly surrounded by windows.

We have compared RPN with the results of Integral Feature Channel (Dollar et al., 2009) and Deformable Part-based Model (DPM) (Girshick et al., 2012) detectors used in previous work (Jampani et al., 2015, Mathias et al., 2016, Rahmani et al., 2017) and found that the RPN performs considerably better.

5. RECTANGULAR FITTING

After labeling by the SRF, the image is post-processed. We employ architectural constraints embodying the following assumptions: The facade objects window, door, balcony and shop have a rectangular shape, roof and wall are divided by a horizontal straight line and windows from a grid structure.

First, we count all vertical and horizontal “changes” between window, door, balcony and shop to wall or other objects and vice-versa. This reduces the search space for object boundaries and from the statistics we derive the grid structure. Other methods delineate the objects based on the objects on the same row and column. We abstain from this, because the height and the width, particularly of windows, can change depending on the viewing angle of the image and further image distortions incurred during the rectification of the image. Our fitting is, thus, based only on the local labeling.

We have defined a minimization function to fit the objects in the rectangle,. Formally, let rectangle R_{x_1,y_1,x_2,y_2} be defined by its upper left (x_1, y_1) and bottom right corner (x_2, y_2). Let our object to be fitted have class label o_c. We then define the optimization problem as follows:

$$\operatorname*{argmin}_{x_1,x_2,y_1,y_2} \sum_{i=x_1}^{x_2} \sum_{y_1}^{y_2} I[L(i,j) \neq o_c] + \\ \sum_{i=x_i-k}^{x_2+k} \sum_{y_1-k}^{y_2+k} I[L(i,j) = o_c \wedge (i,j) \notin R_{x_1,y_1,x_2,y_2}] \quad (4)$$

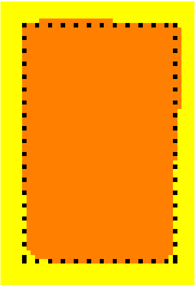


Figure 3. Rectangular Fitting for a door. (a) The orange polygon is generated by the SRF and the dashed line represents the rectangle with minimum loss.

where $I[]$ is the indicator function and k a parameter which is empirically determined. Hypotheses are generated from the statistics of “changes” and the rectangle $R_{x_{1'},y_{1'},x_{2'},y_{2'}}$ with the minimum score is selected.

Intuitively, this minimization function produces rectangles which contain as many pixels of class o_c as possible and try to avoid pixels of other classes (Fig. 3).

To compute the number of pixels in each rectangle, we use an integral image representation (Viola and Jones, 2001). With it, the score of each rectangle can be computed with $O(1)$ time complexity. Compared to other methods such as CRF (Jampani et al., 2015) and DP (Cohen et al., 2014, Cohen et al., 2017), our optimization method is fastest in terms of time complexity and, still, produces the highest quality results (Figs. 4 and 5).

6. EXPERIMENTS

6.1 Datasets

ECP This dataset consists of 104 rectified facade images of buildings from Paris with Hausmannian architecture. All images are labeled according to the seven semantic classes *balcony, door, roof, shop, wall and window* with balconies, doors, windows and shops modeled as rectangles. The dataset was first published by (Teboul et al., 2011) but some annotations were not correct. In 2012 (Martinović et al., 2012) corrected these annotations. In our experiments we use the later dataset.

Graz The dataset contains 50 rectified images from Graz (Riemenschneider et al., 2012) comprising facades of different architectural styles (Classicism, Biedermeier, Historicism and Art Nouveau). Each image is labeled according to the four semantic classes *door, sky, wall and window*

To train the RPN for the ECP dataset we have collected and annotated a larger number of facade images.

6.2 Image Features

The SRF has been trained with similar channels as in (Rahmani et al., 2017), such as RGB and CIELab color, HOG, location information, 17 TextonBoost filter responses and the scores of the RPN. We have removed the features that have a low information gain. We use Extremely Randomized Trees (Geurts et al., 2006), which results in high quality trees as well as fast computation and results of higher quality.

Since the facades are rectified, it is meaningful to add as features the row variance and median of the RGB channels as well as the corresponding deviation from the median. These “statistical” features have a high information gain especially for the Graz dataset for which the content is simpler and where the images do not contain occlusions.

7. EVALUATION

We have evaluated our algorithm on both datasets using 80% of the data for training and 20% for testing. The split of the dataset has been chosen randomly. The SRF for the ECP dataset is trained with patches of size 17×17 and for the Graz dataset we have chosen as patch size 11×11. For the ECP dataset we have evaluated also patches of sizes 15×15 and 19×19 and for the Graz dataset of sizes 13×13 and 15×15, but the results were not significantly different (the absolute difference in accuracy was 0.1%).

The empirical results have been compared with the results for current published work on the datasets (Martinović et al., 2012, Mathias et al., 2016, Jampani et al., 2015, Cohen et al., 2014, Cohen et al., 2017, Rahmani et al., 2017). From Tables 1 and 2 one can see that our method outperforms the other methods. Our algorithm is more than 2% better on the Graz dataset and 0.4% on the ECP dataset than the current state of the art. Additionally, our method is an order of magnitude better than other methods in recognizing doors due to the good cooperation of the RPN and SRF.

We have evaluated our algorithm in four stages: the Baseline (ours^{Base}) evaluates the Structured Random Forest performance without the RPN. For the ECP dataset this leads to weak results for windows and, particularly, doors. Since the content of the Graz dataset images is simpler, the statistical features suffice for it to produce good results.

Incorporation of the RPN features into the SRF (ours^{RPN}) significantly improves the performance. The accuracy for doors on the ECP dataset increases by more than 30% and is better than the current state-of-the-art methods by more than 7%. Additionally, the RPN also improves the accuracy for window by 7%. The windows located on the roof are occluded by balconies due to the viewing angle. This affects our algorithm, which labels the occluded window parts as balcony (Figs. 5 and 6) and,thus, does not achieve an accuracy higher than 80% on the ECP dataset. In the Graz dataset, the RPN also improves the door and window accuracy, but not with the same magnitude as for the ECP dataset.

The optimization step (ours^{O}) does not significantly improve the overall result quantitatively, but it removes noise which positively effects the rectangular fitting.

The rectangular fitting (ours^{RF}) improves the overall accuracy by more than 0.5% for both dataset. It creates high-quality labeled images. The final result is suitable for many applications that need highly precisely delineated facade objects, such as 3D city models.

Finally, we note that the developed pipeline has also its weaknesses. Particularly during learning, the SRF develops a high confidence in the RPN since its features have a high information gain. Thus, when the RPN is wrong, i.e., proposes an object with a high probability which actually does not exists, the SRF is not able to recover (Fig. 6).

8. CONCLUSION AND FUTURE WORK

We have presented a method for facade segmentation which outperforms other state-of-the-art methods in terms of accuracy and quality. The Structured Random Forest, the Region Proposal Network based on a Convolutional Neural Network and the rectangular fitting method constitute a very good combination for facade segmentation. Rectangular model fitting is particularly suitable for this task due to the shape of the facade objects. With the assumption of a grid structure for windows we added a global constraint. Finally we note that our novel approach does not only produce a high quality result, but is also very efficient. We consider to implement and improve it in a way that it can process more than one facade image per second on a normal computer without a significant influence on the accuracy.

REFERENCES

Cohen, A., Martin, O., Yanxi, L. and Pollefeys, M., 2017. Symmetry-aware facade parsing with occlusions. In: *3DV*.

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.

Geurts, P., Ernst, D. and Wehenkel, L., 2006. Extremely randomized trees. *Machine learning* 63(1), pp. 3–42.

Girshick, R. B., Felzenszwalb, P. F. and McAllester, D., 2012. Discriminatively trained deformable part models, release 5.

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.

Class	Results								
	I	II	III	IV	V	ours^{Base}	ours^{RPN}	ours^{O}	ours^{RF}
Sky	91	88	88	90.6	75.7	86	89.9	90.0	**91.3**
Window	60	76	85	80.9	79.2	79	80.6	80.9	**83.7**
Door	41	52	64	63	79.0	91.8	93.7	**94.1**	93.8
Wall	84	95	96	95.8	96	96	96.1	96.3	**96.5**
Overall	78.0	89.6	91.6	91.68	91.1	91.9	92.6	92.8	**93.6**

Table 1. Labeling results on dataset Graz of different methods (I-(Riemenschneider et al., 2012), II-(Cohen et al., 2014), III-(Cohen et al., 2017), IV-(Jampani et al., 2015), V-(Rahmani et al., 2017)), ours^{Base} – Structured Random Forest (SRF) only, ours^{RPN} – SRF with Region Proposal Network (RPN), ours^{O} – SRF with optimization and ours^{RF} – Rectangular Fitting applied to ours ^{O}. Best results given in bold.

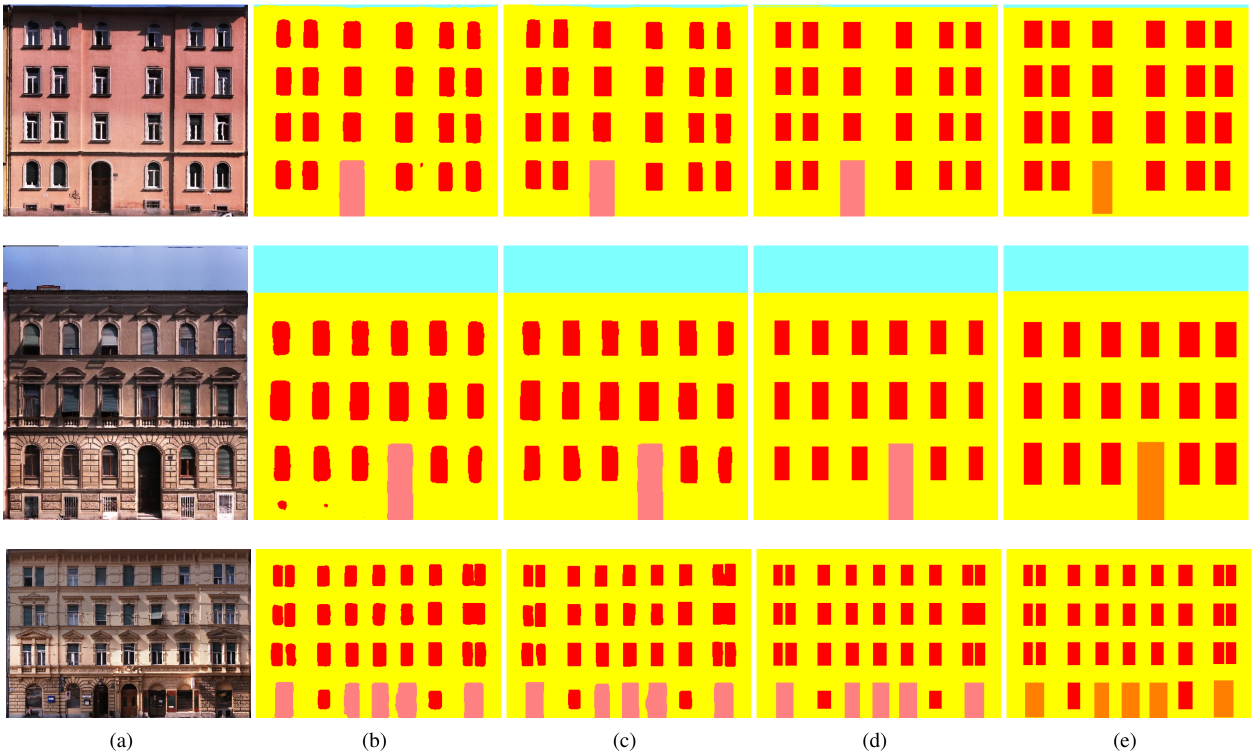
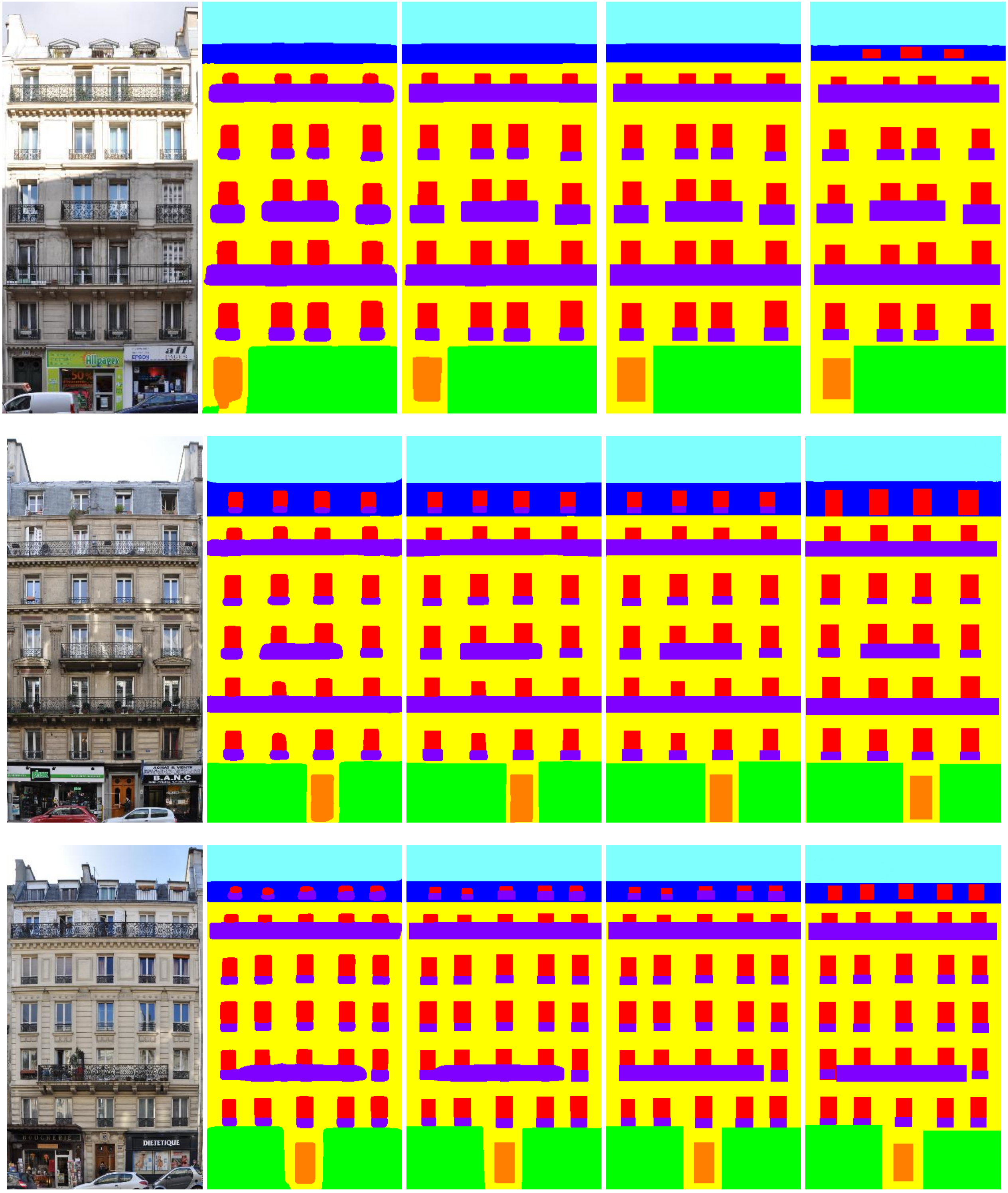


Figure 4. Qualitative results on the Graz dataset. The facade segments are homogeneous and nearly without noise. Column (a) input images, (b) results from Structured Random Forest with Regional Proposal Network, (c) results after 6 iterations of the optimization, (d) results after Rectangular Fitting and (e) ground truth. Object classes: 🟥 - window, 🟧 - door, 🟦 - sky, 🟨 - wall

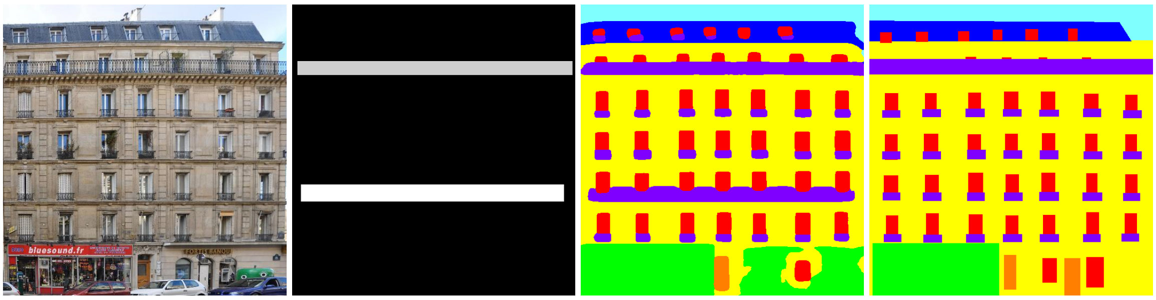
Class	Results								
	I	II	III	IV	V	Ours^{Base}	Ours^{RPN}	Ours^{O}	Ours^{RF}
Door	81.3	79	79.47	79	71	56.6	89.1	89.0	**89.2**
Shop	93.2	94	95.17	96	95	95.3	95.5	96.0	**96.3**
Balcony	89.3	91	86.43	**92**	87	86.4	89	90.0	89.2
Window	82.3	85	80.41	**87**	78	70.4	77.3	78.0	78.6
Wall	92.9	90	91.52	91	89	91.4	92.9	92.6	**93.5**
Sky	**98.2**	97	96.18	97	96	95.3	97.1	97.2	97.2
Roof	89.2	93	91.02	91	79	93.2	**94.9**	94.8	93.0
Average	89.49	89.4	88.60	-	85.22	84.11	90.8	**91.1**	91.0
Overall	91.42	90.82	90.24	91.8	88.02	88.9	91.5	91.6	**92.2**

Table 2. Labeling results on dataset ECP of different methods (I-(Riemenschneider et al., 2012), II-(Cohen et al., 2014), III-(Cohen et al., 2017), IV-(Jampani et al., 2015), V-(Rahmani et al., 2017)). ”ours”, cf. Table 1. Best results given in bold.



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

Figure 5. 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 with Regional Proposal Network, (c) results after 19 iterations of optimization, (d) results after Rectangular fitting, (e) ground truth. Object classes: 🟥 - window, 🟧 - door, 🟪 - balcony, 🟦 - sky, 🟨 - wall, 🟦 - roof



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

Figure 6. This segmentation has a low accuracy since the object detector proposes with high confidence a long (running) balcony in the middle of the facade. Although this long balcony does not exist, the Structured Random Forest (SRF) “trusts” the Region Proposal Network(RPN) output and the SRF produces a wrongly labeled image. Column (a) input image, (b) RPN output for long balcony (c) result from SRF, and (d) ground truth. Object classes: 🟥 - window, 🟧 - door, 🟪 - balcony, 🟦 - sky, 🟨 - wall, 🟦 - roof

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., Gadde, R., Zagoruyko, S., Obozinski, G. and Marlet, R., 2015. A mrf shape prior for facade parsing with occlusions. In: *Computer Vision and Pattern Recognition*, pp. 2820–2828.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp. 1097–1105.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E. and Jackel, L. D., 1990. Handwritten digit recognition with a back-propagation network. In: *Advances in neural information processing systems*, pp. 396–404.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In: *European conference on computer vision*, Springer, pp. 740–755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C., 2016. SSD: Single shot multibox detector. In: *ECCV*.

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. *PCV* 6, 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.

Rahmani, K., Huang, H. and Mayer, H., 2017. Facade segmentation with a structured random forest. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*, pp. 91–99.

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., Bódis-Szomorú, A., Weissenberg, J. and Van Gool, L., 2014. Learning where to classify in multi-view semantic segmentation. In: *European Conference on Computer Vision*, Springer, pp. 516–532.

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.

Schmitz, M. and Mayer, H., 2016. A convolutional network for semantic facade segmentation and interpretation. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 41, pp. 709.

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.

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

Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, IEEE, pp. I–I.