Large Scale Asset Extraction for Urban Images

Lama Affara, Liangliang Nan, Bernard Ghanem, and Peter Wonka

King Abdullah University of Science and Technology (KAUST), Saudi Arabia
lama.affara@kaust.edu.sa, liangliang.nan@gmail.com,
bernard.ghanem@kaust.edu.sa, pwonka@gmail.com

Abstract. Object proposals are currently used for increasing the computational efficiency of object detection. We propose a novel adaptive pipeline for interleaving object proposals with object classification and use it as a formulation for asset detection. We first preprocess the images using a novel and efficient rectification technique. We then employ a particle filter approach to keep track of three priors, which guide proposed samples and get updated using classifier output. Tests performed on over 1000 urban images demonstrate that our rectification method is faster than existing methods without loss in quality, and that our interleaved proposal method outperforms current state-of-the-art. We further demonstrate that other methods can be improved by incorporating our interleaved proposals.

1 Introduction

The goal of our work is to efficiently extract high-quality assets of architectural elements from a large set of urban images. We define an asset as a rectangular image region depicting an architectural element, such as a window, door, ledge, ornament, or even a complete facade. Because these assets might be captured from different viewpoints, their corresponding image regions should be rectified and stored as textures, so they can be subsequently incorporated into applications such as urban modeling or architectural analysis.

We formulate asset extraction as an object detection problem with the following focus. First, we require a fast overall processing time, where we expect each urban image to require at most 1s to be processed by our end-to-end pipeline. We believe that longer processing times are not feasible for practical scenarios where the image dataset is large nor for interactive applications. Second, small low-resolution assets are not interesting for most applications, so we make sure to ignore regions covering only a few pixels. Third, while the fundamental principles of our work are applicable in a wider context, we build a framework best suited for extracting architectural assets.

To tackle this problem, we build on recent work in the area of object proposals. Since processing time is crucial in our application, we assume that a state-of-the-art asset classifier can be learned from a given training set and that it would take more than 1s to exhaustively apply this classifier on all possible image regions to retrieve high-quality assets. In this case, we pose the question:

which image regions are good candidates for the classifier? Recent work on object proposals, namely EdgeBoxes [33] and Geodesic proposals [14], have shown great promise in the field of object detection. In fact, they have become standard methods in many state-of-the-art detection pipelines. Also, we found these two methods to be the most competitive on our urban asset datasets and we therefore use them in our comparisons.

The basic idea of these two approaches (as well as most other object proposal methods) is to find a simple scoring function that can quickly determine whether a candidate region is likely to contain an object or not, thus, reducing the number of negative instances that an object classifier is applied. We would like to extend this idea to interleave object proposals with the classification. In doing so, the proposal method evolves and adapts to the streaming image data. By analyzing the classification result in a previous image or in a particular image region of the current image, we can build insight on what image regions should be considered next as proposals. We realize this idea using a probabilistic sampling strategy akin to a particle filter, where the posterior distribution of the next proposal state (e.g. scale, location, and aspect ratio) is updated with more observations (outputs from the classifier).

Contributions. This work makes three main contributions. **(1)** For the task of urban asset detection, we improve upon state-of-the-art object proposals by using the concept of interleaved proposing and classification. **(2)** We propose a novel rectification algorithm for images with a dominant plane, in general, and urban images, in particular. This method is 18 times faster than previous work. **(3)** We compile and manually annotate a large-scale dataset of urban images, where each facade and window asset are precisely labelled. Our extensive experiments will show that our interleaved proposal technique can outperform state-of-the-art proposal methods on over 1000 facades, as well as, be incorporated into these methods to improve their performance on the same facade scenes.

2 Related Work

Asset extraction relates to techniques ranging from low-level image segmentation to high-level semantic modeling. In this section, we mainly review the work that is closely related to ours, in particular, facade parsing and object proposals.

Facade Parsing. This task aims to decompose facade images into semantically meaningful parts. One key ingredient to facade parsing is the detection of translational symmetry, e.g. Müller et al. [19]. To classify facade elements, many recent papers [4, 20, 17] employ machine learning techniques . By combining random forests with shape grammars, Teboul et al. [25] incorporate reinforcement learning for parsing facades. Cech et al. [1] define it as a maximum aposteriori probability labeling task. By assuming s structured facade can be represented as a rank-one matrix, Yang et al. [30] formulate the problem as a matrix decomposition problem. Riemenschneider et al. [20] utilize low-level classifiers combined

with middle-level object detection for parsing irregular facades. They adapt a variation of the Cocke-Younger-Kasami (CYK) algorithm [22] for the split grammars, and the complexity of the CYK algorithm is significantly reduced by representing the facade as irregular lattices. By combining split grammar and shape priors, Kolinsky et al. [13] enable parsing of occluded facade parts.

In practice, shape grammars are usually manually constructed by experts for style-specific facade structures. To avoid this, Martinović et al. [18] propose a three-layered facade parsing approach that combines supervised learning with object detection and knowledge of the architecture. Borrowing ideas from natural language processing, authors in [17, 28] propose to learn context-free grammars based on Bayesian Model Merging [23]. For efficiency, they exploit the similar idea of representing the facade structures as 2D lattices as in [20]. While some of these facade parsing methods can generate very good results, the computation time is too high for our application.

Object Proposals. In order to extract meaningful facade elements, an alternative approach is to use object detection techniques. However, traditional techniques usually apply sliding window search [10, 15, 9], which is computationally inefficient. In recent years, researchers proposed generic objectness proposals to reduce the large number of candidate windows [21, 31, 27, 7, 2].

By encoding the likelihood of neighboring superpixels into a connectivity graph, Manen et al. [16] generate object proposals as bounding boxes of randomized partial spanning trees. Krähenbühl et al. [14] also use superpixels where critical level sets in geodesic distance transforms are computed. Cheng et al. [2] resize the image windows to a small fixed size and use the norm of the gradients as features to detect object boundaries. This method can generate object proposals at a very high frame rate but at the cost of low recall. Zhao et al. [32] further reduce the number of candidate windows by analyzing the distributions of the locations and sizes of object rectangles. Similar to the work of [2], Edge-Boxes [33] relies on edge information. Specifically, the more the edges contained by a box, the more likely there is an object bounded by this box. For a comprehensive evaluation and an in-depth analysis of object proposal methods, please refer to the recent survey [11].

In this work, we follow the idea of object proposals and extend it to interleave object proposals with the task of classification. We compare our performance to the best state-of-the-art methods in Section 4.2.

3 Our Approach

Our asset extraction framework detects bounding boxes around urban assets using four major components visualized in Fig. 1.

1. **Preprocessing:** Given an input image, we first detect line segments and rectify the image based on the detected most dominant facade. We then extract rectangular superpixels to restrict the search space for assets. (See Section 3.1)

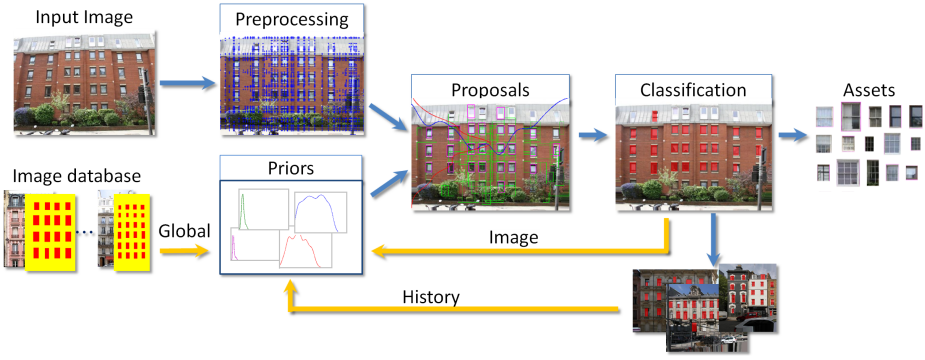


Fig. 1: Overall pipeline. Given an input image, rectification and rectangular superpixels are calculated. Proposals are generated using the estimated global priors as well as the adaptively updated history and image priors using the particle filter strategy. The output of our pipeline is urban assets stored as textures.

2. **Prior Estimation:** We employ a particle filter to estimate prior distribution functions (pdf) for the four bounding box parameters. For each parameter, three pdfs are estimated and updated during detection: global prior, history prior, and image prior. (See Section 3.2)
3. **Object Proposals:** Object proposals are sampled as the evolving particle states and guided by the search space induced by estimated rectangular superpixels. (See Section 3.3)
4. **Asset Extraction:** The proposed objects are classified, and the classifier scores are used to update the priors, thus, guiding the sampling of object proposals. (See Section 3.4)

3.1 Preprocessing

Rectification. The first step in our framework is image rectification. Existing work handles this problem by calculating a homography starting from vanishing points (VP) or using relative scale changes constraints [3]. Such approaches have several drawbacks. First, they only transform vanishing lines to parallel, thus only resolving affine transformations, and more constraints need to be added to recover orthogonality and adequate aspect ratio. Second, slight discrepancies in VP locations significantly affect the final rectification. Wu et al. [29] handle this by a VP refinement technique. This brings us to the third disadvantage: the computational cost of finding the VPs and the added cost of their refinement.

Our approach uses line segments without requiring VP detection. We seek the best homography that vertically aligns the line segments vanishing to top/bottom, and horizontally aligns those vanishing to right/left. To achieve this, we decompose the full transformation $\mathbf{H}$ into a concatenation of two simpler transformations: vertical perspective $\mathbf{H_v}$ and horizontal perspective $\mathbf{H_h}$ depicted in Fig. 2.

$$\mathbf{H} = \mathbf{H_v}\mathbf{H_h} \tag{1}$$

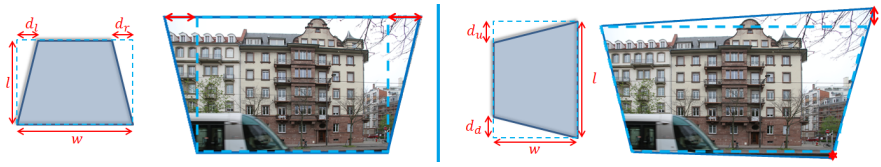


Fig. 2: Vertical perspective (left) and horizontal perspective (right) image transformations with an example horizontal and vertical perspective transformation applied to an urban image (see original image in Fig. 3-left).

We use the following parameters to define the matrices $\mathbf{H_v}$ and $\mathbf{H_h}$: horizontal (d_l, d_r) and vertical (d_u, d_d) shifts of the image corners, as well as, image width w and length l. With these parameters, we can completely model matrices $\mathbf{H_v}$ (refer to Eq. 2) and $\mathbf{H_h}$ (refer to Eq. 3) shown below.

$$\mathbf{H_v} = \begin{bmatrix} 1 + \frac{d_r - d_l}{w} & \frac{-d_l}{l} & d_l \\ 0 & 1 + \frac{d_r - d_l}{w} & 0 \\ 0 & \frac{d_l - d_r}{wl} & 1 \end{bmatrix} \quad (2) \qquad \mathbf{H_h} = \begin{bmatrix} 1 + \frac{d_d - d_u}{l} & 0 & 0 \\ \frac{-d_u}{w} & 1 + \frac{d_d - d_u}{l} & d_u \\ \frac{d_d - d_u}{wl} & 0 & 1 \end{bmatrix} \quad (3)$$

For vertical perspective, we use the following constraint to detect the line segments vanishing to the top/bottom: *Under perspective transformation, the orientations of the originally vertical segments changes linearly with their horizontal position in the image.* Fig. 3, shows a scatter plot of the segment orientations as a function of the position of their midpoints. Using RANSAC line fitting, we filter the lines that agree with this vertical linearity constraint. Now, given a set of k line segments $\{\delta_i\}_{i=1}^{k}$, we find the best parameters that transform these line segments into vertical by minimizing the objective function below.

$$\underset{d_l, d_r}{\text{minimize}} \quad \sum_{i=1}^{k} \left| \frac{h_1 a_i}{h_3 a_i} - \frac{h_1 b_i}{h_3 b_i} \right| \quad (4)$$

where a_i and b_i are the homogeneous coordinates of line segment δ_i's end points. Here, h_1 is the first row of $\mathbf{H_v}$, since we want the x-coordinates of the transformed line segments to be equal, and we divide by h_3, the third row of $\mathbf{H_v}$, since we are dealing with homogeneous coordinates.

For horizontal perspective, we first apply the vertical transformation $\mathbf{H_v}$ to the original line segments and image corners resulting in new line segments δ_i', image width w' and length l'. Then similarly, we filter the transformed line segments that agree with the horizontal linearity constraint and minimize for (d_u, d_d). To solve either optimization problem, we use sequential quadratic programming. Once the four shift parameters are found, the homography is calculated using Eqs. 1, 2, and 3. Fig. 2 shows the result after applying the vertical

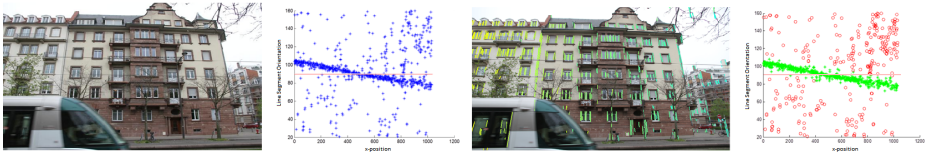


Fig. 3: Segment orientations as a function of the x-position of their midpoints. The line segment orientations vary slightly above 100^0 to slightly below 90^0. Outliers that are discarded by RANSAC are shown in red on the right.

and then the horizontal transformation for the original image shown in Fig. 3.

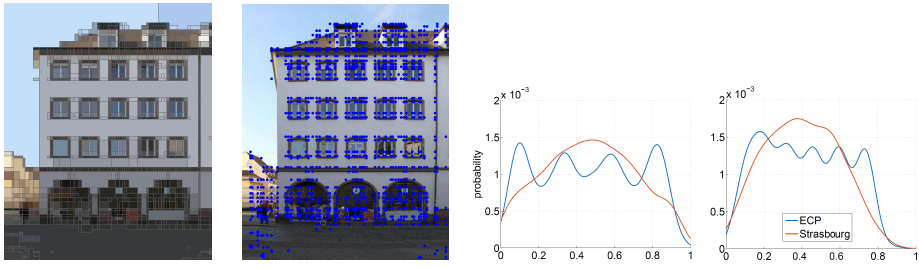
Rectangular Superpixels. Once images are rectified, the most prevalent assets are rectangular elements structured repetitively along horizontal and vertical directions. In fact, currently available urban parsing datasets are based on rectangular boxes of various assets. Based on this observation, we extract superpixels using a rectangular grid at locations where image gradients are large in magnitude. Our goal is a reduced search space $B_s = \{(x_j, y_j, s_j, a_j)\}$ for asset bounding box parameters where (x_j, y_j) is the location, s_j is the scale, and a_j is the aspect ratio. We first sum the image gradient magnitudes along horizontal and vertical directions and use the maxima to form a grid spanning the image. The grid oversegments the image into dense rectangular regions. We then merge color consistent regions to form the final superpixels. The top left corners of these superpixels are used as candidate locations for assets as shown in Fig. 4. We only consider bounding boxes that can be partitioned into superpixels. Finally, loose thresholds on window sizes and aspect ratios are used to eliminate extremely elongated and/or enlarged windows. We constrain scale to be between 0.05 and 0.2 and the aspect ratio to be between 0.25 and 4 in our implementation.

3.2 Prior Incorporation

The extracted search space comprises a large set of object proposals, a subset of which needs to be evaluated by the classifier. We determine this subset adaptively by a particle filter sampling technique. Our approach is adapted to the dataset at hand by estimating the prior probabilities of the parameters and sampling using the joint probabilities as weights. During our framework, we keep track of 12 prior distributions: 3 distributions for each of the 4 parameters.

1. **Global Prior** p^g is estimated from the training database. This prior gives us general information about the parameters. Window assets for example have mean aspect ratio of less than 1 while for balconies it is greater than 1.

2. **History Prior** p^h is updated by the detection history within an image dataset. Facade structures differ from city to city, and learning dataset-specific parameters helps in sampling based on the learned structure. Fig.



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

Fig. 4: A visualization of superpixels as regions (a) and superpixels' top left corners (b). Estimated history priors for x-locations (c) and y-locations (d) in the ECP and Strasbourg datasets. By looking at the distributions, one can note the difference in the structure of the two datasets.

4 shows the difference in the learned history prior for locations of window assets in the ECP (taken in Paris) and the Strasbourg datasets.

3. **Image Prior** p^i is updated by the detection scores in the current image. This prior guides the parameters according to the detected windows in the same image. As soon as new positive detections arise, the prior increases the weights for its parameters. This suits the problem of asset detection very well, since assets are usually repeated along aligned locations in the same image and with similar sizes.

Prior Update as a Particle Filter. We estimate and update the global, history, and image prior distributions used in generating the object proposals from a set of given bounding box samples. The global prior is estimated only once from the ground truth bounding boxes in the training set using kernel density estimation. The history prior is also estimated using kernel density estimation, but from the classified bounding boxes in the dataset. As for the image prior, we update it by casting it in a particle filter framework.

Particle filtering is a general Monte Carlo (sampling) technique used to estimate the state of a system that evolves over time. The general filtering problem involves the estimation of the posterior distribution of state variables $\mathbf{x}_t$ at time t given all observations $\mathbf{z}_{1:t}$ up to time t. The computation of the distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ can be done recursively in two major steps: (1) the prediction step which is computed using the previous state and all previous observations, and (2) the update step in which the distribution is updated with the new observation $\mathbf{z}_t$ using Bayes' rule.

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1} \tag{5}$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) \tag{6}$$

In general, Eqs. 5 and 6 are computationally intractable, hence particle filtering can be viewed as an approximate method for system state estimation, especially when the posterior state distribution is not Gaussian. Therefore, we estimate the posterior distribution at time t with a weighted set of samples $\{(\mathbf{x}_t^i, w_t^i), i = 1...N\}$, also called particles, and recursively update these particles to obtain an approximation to the posterior distribution at the next time step. The particle filter prediction (see Eq. 7) and update (see Eq. 8) steps are combined to assign an importance weight for each sample.

$$\mathbf{x}_t^i \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^i) \tag{7}$$

$$w_t^i = w_{t-1}^i p(\mathbf{z}_t|\mathbf{x}_t^i) \tag{8}$$

In our case, the state $\mathbf{x}_t \in \mathbb{R}^4$ represents the four parameters defining an object proposal (i.e. location, size, and aspect ratio). In other words, the particles are sampled bounding boxes in the image, while the observations are manifestations of particles in an image, namely image patches enclosed inside the particle's bounding box. The prediction step gives rise to the transition model of the particle filter and defines how the particles evolve in each iteration. Unlike many other use cases of particle filtering where the transition model is simplistic and assumed to be zero-mean Gaussian, we exploit the horizontally and vertically repeating patterns, which govern the spatial distribution of urban assets in facade images, to define this spatial transition model. This probabilistic model is shown in Eq. 9, where $\mathbf{r} = (r_x, r_y, 0, 0)$ denotes the estimated horizontal and vertical repetition intervals, $\mathbf{k} = (k_x, k_y, 0, 0)$ is a random variable which denotes the sampled number of repetitions, $\odot$ is the element-wise multiplication operator, and $\nu \sim N(\mathbf{0}, \Sigma)$ is additive Gaussian noise. Moreover, the update step gives rise to the observation model $p(\mathbf{z}_t|\mathbf{x}_t^i)$ and is taken to be the normalized score of the asset classifier when it is applied to the observed image patch $\mathbf{z}_t$.

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{k}_{t-1} \odot \mathbf{r}_{t-1} + \nu \tag{9}$$

By iterating through the aforementioned prediction and update steps, particles that manifest themselves as assets in an image maintain a higher weight than those that are not, which in turn biases the sampling strategy to focus more on the former particles in the next prediction iteration. As such, we use the asset content of an image to guide the particle sampling and, in turn, gradually evolve the image prior. In Fig. 5, we show the evolution of three particle filters on an example facade image. We first initialize the system state by probabilistically sampling bounding boxes using the learned priors. Starting from the initial state in the first iteration, horizontal and vertical repetitions are estimated and the particles evolve according to the previously described prediction step to produce the bounding boxes shown in Iteration 2. The particles keep evolving until particle discovery converges (i.e. when previously covered states are repeated).

3.3 Guided Object Proposals

The particle filtering method can be viewed as an interleaving process between object proposals and classification. The output of the classification at each time

Fig. 5: The evolution of the particle filter across an image. At the first iteration, particles are sampled around three initial states. The bounding boxes show the sampled particles with the weights represented by the color map on the left. Particles with higher weights are used more often in subsequent iterations.

step is fed back as weights to the particles, which are sampled based on the updated weights. The sampled N particles (bounding boxes) are output as object proposals and they are guided by the image and history priors. To guarantee sample diversity at each time step, we combine the three priors using a user-defined weighted sum as shown in Eq. 10 where we use $w_g = 0.3$, $w_h = 0.3$ and $w_i = 0.4$ in our implementation and we start with $N = 60$ particles.

$$p = w_g p^g + w_h p^h + w_i p^i \tag{10}$$

The repetition intervals are estimated at each time step by a voting scheme using the absolute horizontal and vertical weighted distances calculated from the sampled particles. The number of horizontal k_x and vertical repetitions k_y for each particle are sampled from a range $[k_{min},k_{max}]$, using the prior probability of the x-location and y-location generated by shifting the particle by a $(r_x k_x, r_y k_y)$ translation. In our implementation, we use the range [-2,2]. Since we use multiple iterations, it is still possible to sample very large grids.

The image and history priors characterize different time scales of the observation history, and thus need to be updated at different time steps. We update the image prior k_i times in the same image. The history prior however is updated every k_h images. Small k_i and k_h means we are getting more feedback from the classifier, but at the cost of higher computation time. Increasing k_i and k_h however decreases classifier feedback which takes us back to traditional proposal methods. In our implementation, we choose $k_i = \frac{\#proposals}{20}$ and $k_h = 24$.

3.4 Asset Classification

Given a set of object proposals, a linear SVM classifier is used to classify which asset class each proposal belongs to, if any. We use aggregate channel features [6] formed of 6 gradient orientations, 1 gradient magnitude, and 3 L-ab color channels to compute the feature descriptors. Inspired by the work of [6], instead of computing the exact features after resizing each object proposal patch to the

Algorithm 1 Guided Asset Extraction

```
Train Classifier
Estimate Global Prior
for each image I_0 ∈ Dataset do
    I =Rectify(I_0)
    B_s =Rectangular Superpixels(I)
    Sample Particles x_1^1, ..., x_1^N
    Compute Weights w_1^1, ..., w_1^N
    for t = 2 to T do
        Predict States x_t^1, ..., x_t^N
        Update Weights w_t^1, ..., w_t^N
        Resample Particles
        #proposals + = #particles
        if processed k_i proposals then
            Update Image Prior
        end if
    end for
    if processed k_h images then
        Update History Prior
    end if
end for
```

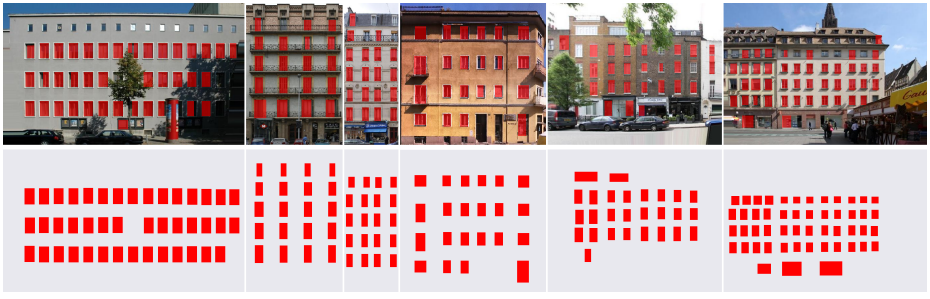


Fig. 6: Final detections on random images from the eTrims, CMP, ECP, Graz, London, and Strasbourg datasets (from left to right) with corresponding ground truth (bottom). Both our algorithm and ground truth are configured to ignore small windows.

classifier model size, we compute the channels on the full image and resize the channels. This leads to 3× speedup without much loss in final precision.

The classifier is applied at each update step on all the sampled object proposals. Its output is taken as final detection score for the currently classified proposals and is used in the particle filtering weight assignment step. The overall pipeline of our framework is described in Algorithm 1.

4 Results

We implement the algorithms described in the paper in MATLAB using the Parallel Computing Toolbox when possible and we use an Intel 3.1GHz processor machine for testing. We used the Piotr toolbox [5] for superpixel estimation and feature extraction and the LibLinear package [8] for the SVM classifier. **Datasets.** We use the following datasets for evaluation: GRAZ50 [20], ECP [24], eTrims [12], CMP [26] and two new datasets for *London* and *Strasbourg* that we manually compiled comprising 800 images taken in the wild. The total dataset size is 1392 images. We developed a user interface for rectifying and labeling

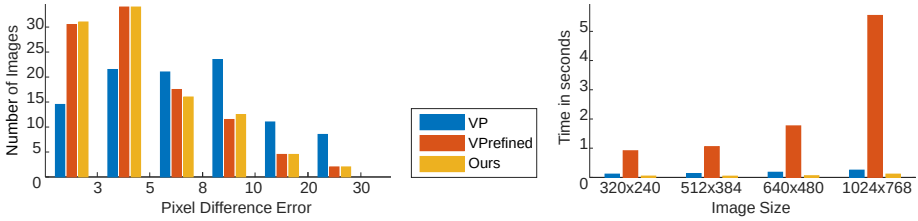


Fig. 7: The rectification accuracy for a dataset of 100 images (left). For each image we compute the pixel error compared to ground truth and report the results as a histogram. On the right we show computation time for images of different sizes.

facade images to obtain asset bounding boxes (e.g. for windows). The labeling of ECP is pixel based, and does not label window parts that are occluded by balconies as windows. Thus, we relabeled it to include the complete windows. We also annotated the ground truth for London and Strasbourg, by marking bounding boxes around urban assets (See Fig. 6 bottom).

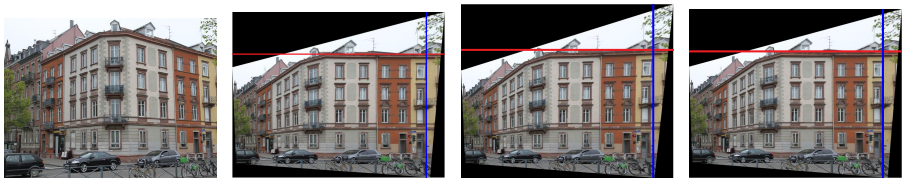
We give an overview of the results below and more details in the **supplementary material**. We show an evaluation of the homography estimation, the object proposal recall rates, and PR curves for the final detection output. The goal of our asset extraction pipeline is to extract as many useful assets per time unit as possible, while maintaining a reasonable precision. Our results will show that our framework is better suited to serve this goal than previous work.

4.1 Homography Estimation

We compare our homography estimation method against two variants of the method by Wu et al. [29]: *VP*, a faster version using calculated VPs before refinement, and *VP-refined*, which uses the extracted VPs after refinement. To quantitatively evaluate the accuracy, we manually annotate a dataset of 100 images. Ground truth is depicted as lines that should be transformed to horizontal/vertical after rectification. The inconsistency of applied rectification is calculated as the average pixel difference error of transformed lines compared to the average correct line. Fig. 7-left shows that the VP method is significantly less accurate than the other two approaches, and that our approach is comparable to that of VP-refined. We also evaluate the average rectification speed. Fig. 7-right shows that our rectification method is faster than both approaches especially as the images grow in size. For 640×480 images, our method is 2 times faster than VP and 18 times faster when VP refinement is applied. Fig. 8 shows a comparison of the final output of the three methods on a sample image.

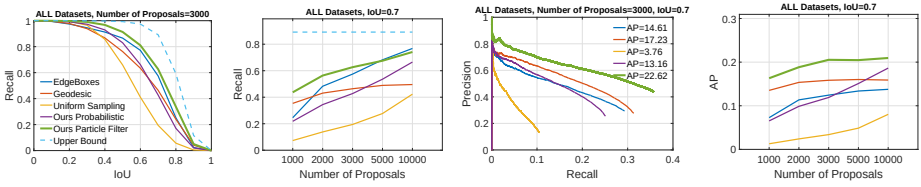
4.2 Object Proposals

We follow common practice to evaluate the quality of object proposals based on recall of ground truth annotations. The recall rate defines the fraction of ground



(a) Original Image (b) Ours (c) VP (d) VP-refined

Fig. 8: Rectification applied on original image in (a) using three different methods. Notice in the output of the VP method (c), alignment artifacts are present in the right side of the image, which are fixed after refinement in (d).

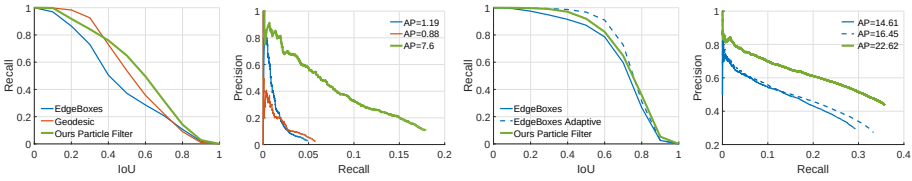


(a) (b) (c) (d)

Fig. 9: Object proposals recall and detection results.

truth retrieved with respect to an intersection over union (IoU) threshold. We use two metrics to evaluate the quality of proposal methods: (1) recall as the IoU threshold is varied for a fixed number of proposals, and (2) recall as the number of proposals is varied for a fixed IoU threshold. We evaluate the recall rates on the urban dataset described earlier. We compare against two existing proposal methods EdgeBoxes [33] and Geodesic [14]. We also evaluate the following proposal strategies (1) *Uniform Sampling* where we sample uniformly from the reduced search space, (2) *Ours Probabilistic* where we sample based on global prior only, (3) *Ours Particle Filter* where we sample using our particle filter approach with interleaving object proposals and classification, and (4) *Upper Bound* which shows the recall upper bound due to the reduced search space induced by our superpixel pre-process. Based on visual inspection, we choose an IoU threshold of 0.7 as the most reasonable in practice and fix number of proposals to 3000. For EdgeBoxes and Geodesic, we use the default parameters and models provided in the authors' implementations.

As shown in Fig. 9a and 9b, our approach provides the highest overall recall, competing with EdgeBoxes for more than 5000 proposals. The particle filter transition model together with the interleaving strategy give rise to an adaptive search space and thus high recall rates. In addition, the structure of urban assets agrees well with the EdgeBoxes scoring function which uses object boundaries estimates as features. Geodesic uses a segmentation scheme for proposing objects, and increasing the number of proposals doesn't improve their recall much.



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

Fig. 10: Recall and PR curves using 3000 proposals on the balconies dataset (left) and for our prior combined with EdgeBoxes (right).

Fig. 10a also shows the recall rate of our method compared to EdgeBoxes and Geodesic on the ECP and CMP datasets using balconies assets.

These recall rates evaluate the quantity rather than the quality of object proposals as they do not include precision or accuracy of returned bounding boxes. Our main concern in this pipeline is the final detection output and an evaluation of the precision of retrieved detections after applying the classifier on top of proposed bounding boxes. EdgeBoxes has high recall for ground truth windows, but at the cost of an increase in the number of false positives as we will show in the next section.

4.3 Asset Classification

We trained a Linear SVM classifier on a separate urban images dataset consisting of 3000 positive window examples from different architectural types, and randomly sampled negative examples making sure they have low overlap with the ground truth. We use two metrics to evaluate the detection quality of the proposals: (1) PR curves which show the precision of the classifier at different recall rates for a fixed number of proposals and IoU threshold, and (2) average precision (AP) as the number of proposals vary for a fixed IoU threshold. We apply the same classifier on all shown proposal methods.

As shown in Fig. 9c and 9d, our method has the highest AP, while that of EdgeBoxes gets lower than Geodesic after applying the classifier on top of the proposals. The AP scores of the probabilistic method increases with the increase in number of proposals but is still below our particle filter approach. This shows that our interleaving strategy performs well by capturing the good proposals earlier. We also show in Fig. 10b the AP scores for balconies assets.

To further show how the priors help get better proposals, we apply the prior weights as a scoring function for the exhaustive space of proposals retrieved by EdgeBoxes. Fig. 10 shows how adding the priors improves the recall and thus AP of EdgeBoxes. In general, adding our adaptive priors to EdgeBoxes consistently extends the recall of EdgeBoxes, while not affecting the precision much. This means that the priors are reordering the proposals in such a way that better proposals are getting higher score.

Table 1: Running time (in seconds) to generate 3000 proposals.

Method	Finding Proposals	Asset Classification	Total
Geodesic	0.18	4.32	4.5
EdgeBoxes	0.1	1.26	1.36
Ours Probabilistic	**0.09**	**0.24**	**0.33**
Ours Adaptive	-	-	**1.1**

4.4 Computational Cost

In this section, we evaluate the full pipeline comparing existing work for object proposals with ours. Table 1 shows the average running time to generate 3000 proposals from 640×480 resolution images in the urban dataset, described earlier. For our adaptive method, separating the computation time for object proposals and classification is difficult as they are interleaved steps, while for the other methods, the two steps are applied separately. As shown in Table 1, our probabilistic method gives the highest computation speed while the adaptive method comes next. The primary cause for increase in computation time is the prior update step. Since our MATLAB implementation is not optimized for speed, we would expect a significant speedup from a C++ implementation. Note that the running time for asset classification using Geodesic and EdgeBoxes proposals is larger. This is related to the quality of proposals returned by these methods (i.e. when providing more proposals that differ much in size/aspect ratio from that of the model, the feature extraction step gets slower because the proposals needs to be resized to fit the classification model). Our guided priors however make sure that we get consistent window sizes and aspect ratios, which give us the advantage of higher speeds. Fig. 6 shows final detection results using our adaptive method on example images from each dataset.

5 Conclusion and Future Work

In this paper, we propose an efficient and effective framework for asset extraction in an urban image dataset. Our main contributions are the interleaving of object proposals with classification to improve overall retrieval results and a new and fast image rectification method. Extensive experiments show that using such an adaptive approach to detect urban assets helps build insight into how to guide proposed image regions, especially in the presence of multiple similar objects in the same image or across images. In future work, we would like to extend our framework to the task of asset extraction in urban videos.

Acknowledgement. This work was supported by the KAUST Office of Sponsored Research (OSR) under Award No. OCRF-2014-CGR3-62140401, and the Visual Computing Center at KAUST.

References

1. Cech, J., Sara, R.: Windowpane detection based on maximum aposteriori probability labeling. In: IWCIA Special Track on Applications. pp. 3–11 (2008)
2. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: Bing: Binarized normed gradients for objectness estimation at 300fps. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2014)
3. Chum, O., Matas, J.: Planar affine rectification from change of scale. In: Proceedings of Asian Conference on Computer Vision. Springer (2011)
4. Dai, D., Prasad, M., Schmitt, G., Van Gool, L.: Learning domain knowledge for facade labelling. In: Proceedings of European Conference on Computer Vision. Springer (2012)
5. Dollár, P.: Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html
6. Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(8) (2014)
7. Endres, I., Hoiem, D.: Category-independent object proposals with diverse ranking. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(2) (2014)
8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of Machine Learning Research 9 (2008)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9) (2010)
10. Gao, D., Mahadevan, V., Vasconcelos, N.: On the plausibility of the discriminant center-surround hypothesis for visual saliency. Journal of vision 8(7) (2008)
11. Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? IEEE Transactions on Pattern Analysis and Machine Intelligence 38(4) (2016)
12. Korc, F., Förstner, W.: etrims image database for interpreting images of man-made scenes. Dept. of Photogrammetry, University of Bonn, Tech. Rep. (2009)
13. Kozinski, M., Gadde, R., Zagoruyko, S., Obozinski, G., Marlet, R.: A mrf shape prior for facade parsing with occlusions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2015)
14. Krähenbühl, P., Koltun, V.: Geodesic object proposals. In: Proceedings of European Conference on Computer Vision. Springer (2014)
15. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2008)
16. Manen, S., Guillaumin, M., Van Gool, L.: Prime object proposals with randomized prim's algorithm. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE (2013)
17. Martinović, A., Van Gool, L.: Bayesian grammar learning for inverse procedural modeling. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2013)
18. Martinović, A., Mathias, M., Weissenberg, J., Van Gool, L.: A three-layered approach to facade parsing. In: Proceedings of European Conference on Computer Vision (2012)
19. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. vol. 26. ACM (2007)

20. Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H.: Irregular lattices for complex shape grammar facade parsing. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2012)
21. Van de Sande, K.E., Uijlings, J.R., Gevers, T., Smeulders, A.W.: Segmentation as selective search for object recognition. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE (2011)
22. Schlesinger, M.I., Hlavac, V.: Ten lectures on statistical and structural pattern recognition, vol. 24. Springer Science & Business Media (2002)
23. Stolcke, A.: Bayesian learning of probabilistic language models. Ph.D. thesis, University of California, Berkeley (1994)
24. Teboul, O.: Ecole centrale paris facades database. URL: http://vision.mas.ecp.fr/Personnel/teboul/data.php
25. Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Parsing facades with shape grammars and reinforcement learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(7) (2013)
26. Tyleček, R., Šára, R.: Spatial pattern templates for recognition of objects with regular structure. In: Pattern Recognition (2013)
27. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International Journal of Computer Vision 104(2) (2013)
28. Weissenberg, J., Riemenschneider, H., Prasad, M., Van Gool, L.: Is there a procedural logic to architecture? In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2013)
29. Wu, C., Frahm, J.M., Pollefeys, M.: Detecting large repetitive structures with salient boundaries. In: Proceedings of European Conference on Computer Vision, vol. 6312 (2010)
30. Yang, C., Han, T., Quan, L., Tai, C.L.: Parsing facade with rank-one approximation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2012)
31. Zhang, Z., Warrell, J., Torr, P.H.: Proposal generation for object detection using cascaded ranking svms. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2011)
32. Zhao, Q., Liu, Z., Yin, B.: Cracking bing and beyond. In: Proceedings of British Machine Vision Conference (2014)
33. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: Proceedings of European Conference on Computer Vision. Springer (2014)