

Tiling of Ortho-Rectified Facade Images

Przemyslaw Musialski* Meinrad Recheis* Stefan Maierhofer* Peter Wonka† Werner Purgathofer*‡
*VRVis Research Center Vienna †Arizona State University ‡Vienna University of Technology

Abstract

Typical building facades consist of regular structures such as windows arranged in a predominantly grid-like manner. We propose a method that handles precisely such facades and assumes that there must be horizontal and vertical repetitions of similar patterns. Using a Monte Carlo sampling approach, this method is able to segment repetitive patterns on orthogonal images along the axes even if the pattern is partially occluded. Additionally, it is very fast and can be used as a preprocessing step for finer segmentation stages.

CR Categories: I.4.6 [Computing Methodologies]: Image Processing And Computer Vision—Segmentation I.4.10 [Computing Methodologies]: Image Processing And Computer Vision—Image Representation

Keywords: image processing, image segmentation, image-based urban reconstruction

1 Introduction

Urban reconstruction is currently undergoing intensive research in the Computer Graphics and Vision communities. One of the still challenging tasks is the recognition and reconstruction of facade details such as windows and ornaments. These are considered key elements of realistic representations of urban environments. In this context, the windows of typical buildings can be seen as patterns that occur multiple times within a rather regular arrangement. Considering a building’s facade on a frontal and orthogonal image, the search for the dominant features can be restricted to only the axis-aligned horizontal and vertical directions.

Our contribution is a method that processes the horizontal and the vertical directions of a rectified frontal facade image independently and delivers a grid of axis-aligned splitting lines. These lines delineate image into regions of high horizontal or vertical translational symmetry. Along these lines, the image can be divided into single repetitive instances. Our method is robust with respect to noise, discontinuities and partial occlusions up to a certain threshold. Moreover, running time is in the order of only a few seconds on mainstream consumer hardware.

In the next section, we give a brief overview of approaches aiming at similar goals, while in section 3 we describe the details of our approach and finally in sections 4 and 5 we present results of our method.

*pm,recheis,sm@vrvis.at

†pwonka@gmail.com

‡wp@cg.tuwien.ac.at



Figure 1: The red lines indicate the grid, which has been detected on the facade. The proposed algorithm is robust to obstacles such as different illumination or reflections in the windows (best seen in color).

2 Related Work

Many of the earlier methods are based on multiple views. Wang *et al.* [Wang and Hanson 2001] merge multiple views in aerial images to reduce shadows and occlusion and detect windows in the blurry results using oriented region growing (ORG). Wang *et al.* [Wang and Hanson 2001] also combine multiple views to eliminate occlusions and use disparity for depth estimation. Then they apply ORG to segment out the windows and a periodic pattern fixing algorithm extrapolates the missing windows. Schindler and Bauer [Schindler and Bauer 2003] detect windows by utilizing depth information reconstructed from multiple views and infer window geometry by template matching on the resulting dense point clouds. Dick *et al.* [Dick *et al.* 2004] reconstruct building geometry from multiple overlapping images and use wavelet decomposition to find windows that have strong horizontal and vertical features. Tsai *et al.* [Tsai *et al.* 2006] make use of color to detect occlusions from vegetation and apply morphological processing to detect the regular structures which they use to repair occluded parts. Brenner and Ripperda [Brenner and Ripperda 2006] apply RJMCMC to create a hierarchical model using grammar rules. Their method is based on both rectified images and laser scan data of facades.

The mentioned methods use either multiple images or images and laser scan data. Recently some methods have been developed which need only a single image of a facade. Lee and Nevatia [Lee and Nevatia 2004] try to find windows based on marginal edge pixel distributions, which provide hypothesis window templates which are then tuned towards the image evidence. Alegre and Dellaert [Alegre and Dellaert 2004] proposed a segmentation of facade images by applying a hierarchical context free grammar. They use Markov Random Fields to tune parameters of rules for a hierarchical context free grammar modeling the facade. Mayer and Reznik [Mayer

and Reznik 2006] detect windows based on an implicit shape model that has been trained on template images. Another similar approach using machine learning and Haar wavelet decomposition has been proposed by Ali *et al.* [Ali *et al.* 2007]. Korah and Rasmussen [Korah and Rasmussen 2007] employ rectangle detection and MCMC to find regular grids of windows in a MRF network. Müller *et al.* [Müller *et al.* 2007] generate a hierarchical procedural model from the image based on translational similarity information measured by mutual information. While their method requires limited user interaction Van Gool *et al.* [Van Gool *et al.* 2007] proposed a method that uses exactly the same approach for images with little perspective but is able to reconstruct building facades automatically for images with strong perspective distortion. Xiao *et al.* proposes first a semi-automatic [Xiao *et al.* 2008] and further an automatic method [Xiao *et al.* 2009] for facade reconstruction. Cech and Sara [Cech and Sara 2008] segment out windows from facades using a MRF based on the fact that windows are strictly axis aligned in almost all cases. Musialski *et al.* [Musialski *et al.* 2009] uses similar sampling as our to detect tiles and applies filtering to repair the facade image.

Apart from these highly specialized methods for building reconstruction there are other related but more general works. Bailey [Bailey 1997] shows that it is possible to detect repetitive image patterns by self-filtering in the frequency domain. Hsu *et al.* [Hsu *et al.* 2001] use wavelet decomposition of the autocorrelation surface to detect image regularities. Liu *et al.* [Liu *et al.* 2004] detect crystallographic groups on repetitive image patterns using a dominant peak extraction method from the autocorrelation surface. Turina *et al.* [Turina *et al.* 2001] detect repetitive patterns on planar surfaces under perspective skew using Hough transforms and application of various grouping strategies. Han and Zhu [Han and Zhu 2009] detect regular rectangular structures in photographs of arbitrary scenes. Their approach combines bottom-up and top-down image interpretation by selecting out of many possible detected candidate rectangles using an attribute grammar. Boiman *et al.* [Boiman and Irani 2007] detect irregularities in images and Shechtman and Irani [Shechtman and Irani 2007] use a similar approach to detect local self similarities. From these they generate robust feature descriptors which they combine to recognizable global ensembles. Furthermore, there exist approaches to detect translational and another symmetry: i.e. Loy *et al.* [Loy and Eklundh 2006] aims in detection of symmetry of particular features in one image and Mitra and Pauly *et al.* [Mitra *et al.* 2006; Pauly *et al.* 2008] introduces an approach to detect symmetric structures in 3D geometry.

Our method is comparable to the firstly mentioned group of single-view approaches which aim at processing a rather specialized task.

3 Recognition of Repeating Patterns

The main idea behind the proposed method is to exploit the inherently repetitive nature of almost all facade elements in order to identify facade tiles, locate them and finally partition the facade image into tiles. The approach to use only the similarity as segmentation criterion arose from the challenge of segmenting typical Art Nouveau facades, which are common in many European cities. Decorated with stucco elements distributed in a relatively unpredictable manner, such facades are particularly challenging to model-based feature detection approaches. Moreover, facades of this category contain many fine grained details and are thus very difficult to model or reconstruct automatically.

In this work translational symmetries are used to identify repetitive features and segment the facades into tiles accordingly. The algorithm takes as input a single orthogonalized view of a facade. The

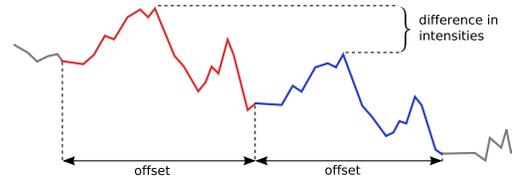


Figure 2: Example of a repetitive pattern in 1D with a highly similar but not identical instance. Relative differences in signal intensities between instances of the pattern should not influence the detection algorithm. An appropriate similarity measure must be applied that is insensitive to the overall intensity level of the region.

output is an orthogonal grid that defines a segmentation of the facade image into repetitive tiles. The algorithm itself is subdivided in the following stages:

Search for dominant repetitive patterns. To identify the relevant repetitive regions of a facade image (e.g., floors or windows) it is necessary to search for similar image regions. This is done by comparing small image regions on multiple resolutions of the image for similarity. Because comparing every pair of potentially corresponding image regions is computationally prohibitive, a Monte Carlo importance sampling strategy is applied to collect statistical evidence about any translational similarities. To extract these relevant patterns out of all the collected evidence the representative offsets are sorted into a histogram where large patterns result in large peaks. These are then extracted by Mean Shift clustering [Comaniciu and Meer 2002]. The result of this stage are offsets in pixels that relate directly to the prevailing repetitive patterns in the image.

Localization and segmentation of the identified patterns. The offsets computed in the previous step convey the size of important repeating patterns but there is no information about their location in the image. In order to determine these locations the image has to be sampled regularly to test the image’s similarity response for a given offset at a given location. Again, efficient randomized multi-resolution sampling approximates a costly per-pixel analysis of the image. The computed similarity curves for every offset are the input to the next stage. Finally the image is partitioned respectively into regions with and without repetitive patterns. For the regions that exhibit repetitive patterns, the most dominant pattern is selected and its offset is taken into account in the splitting process. As a result, the facade is divided into floors and individual window tiles, which can be processed by further algorithms.

3.1 Search for dominant repetitive patterns

A closer look at the typical structure of facades helps to understand which image patterns are relevant for window detection. Most facades feature many windows of the same size and similar appearance. The arrangement of windows is almost always the same for the floors of the same facade. Common exceptions to this rule are usually the first floors which are irregular or different from the others. If we consider a sequence of axis-aligned pixels as a function of the intensities, we notice certain regular repetitions in the signal (Fig. 2). These repetitions are coherent over multiple adjacent pixel lines of the image.

A **repetitive pattern** on a spatial signal is defined in terms of local self-similarities in a 1D signal or 2D image. It is characterized

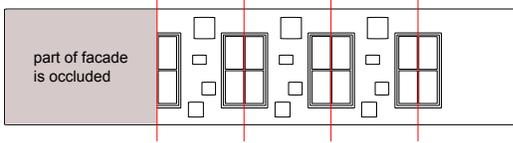


Figure 3: Without a priori knowledge about the signal, it is not possible to evaluate the correctness of a split. In this case half of the first window has been occluded, causing a shift in the start of the repetitive pattern.

by its *offset*, the smallest distance to the next most similar recurrence of certain distinguishable features in the original sequence of the pattern. We call this a *repetitive instance* (see Figure 2). The same image features that are very important for human vision such as edges and corners are most important for our repetitive pattern detection algorithm.

To define the border of a repetitive pattern we assume that the pattern begins at the first distinctive feature (i.e., edge) that is similar to the signal at the characteristic offset and ends as soon as the signal starts to differ too much from the original instance. We constrain the input images to complete pictures of a facade, such that it is impossible (except in case of occlusions like in Figure 3) for a pattern to start in the middle of a window. The bounds of a repetitive pattern are not sharp and have to be defined by a similarity threshold. With such a threshold, non-repetitive regions can be distinguished from pattern regions.

A difficult problem for image segmentation based on repetitive patterns is the handling of overlapping patterns. To demonstrate the problem, consider the facade image in Figure 4. There are two concurring segmentations based on either the one pattern's offset or the other's. A solution to this problem, which is adopted in this paper, is to exclude some of the detected patterns according to a priori knowledge or image area constraints.

Similarity measure. To measure the similarity of image regions we need a robust operator that is suitable for images of repeated real-world objects that can exhibit a large range of defects. In order to compare positions with varying intensities, we compute the normalized cross correlation coefficient (NCC), where we subtract the mean of the intensities $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ of each patch \mathbf{x} and \mathbf{y} and normalize the vectors, respectively:

$$\text{ncc}(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{y} - \bar{\mathbf{y}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \|\mathbf{y} - \bar{\mathbf{y}}\|}. \quad (1)$$

where $\|\cdot\|$ is the Euclidian Norm. The size of the respective vectors \mathbf{x} and \mathbf{y} is equal and is called *window size* further on.

When measuring local similarities, the *window size* is an important parameter to consider with respect to performance and robustness. The cross correlation of small windows like 3×3 or 5×5 pixels can be computed very fast. Larger window sizes, like 63×63 or 127×127 , are very expensive to compute due to the computational complexity of cross correlation which is quadratic in the size of the compared image regions. On the other hand, the quality and robustness of the similarity measure for two image regions increases with larger windows.

When measuring patterns, the size of the pattern relative to the size of the measurement window is very important. If it is too small or too large compared to the measurement window, one will obtain ambiguous results (Fig. 6). Rather than increasing the patch size to improve the robustness of the measure, a very efficient way is

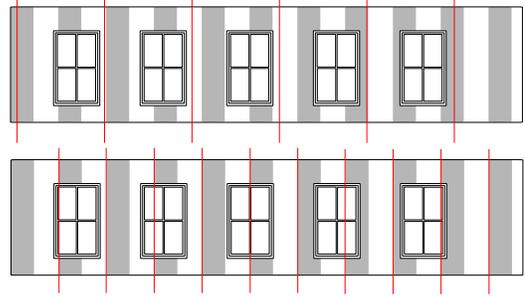


Figure 4: Two overlapping repetitive patterns and their corresponding splitting lines. There are often overlapping patterns, especially in Art Nouveau facades that feature a great deal of decor.

to combine the results of measurements on different scale levels of an image pyramid. This idea has been successfully used in many texture synthesis algorithms. It is computed by subsequently scaling the image with the factor s (in our case we use $s = \frac{1}{2}$ and cubic down-sampling):

$$\zeta(\mathbf{x}, \mathbf{y}) = \frac{1}{N_s} \sum_k^{N_s} \text{ncc}_k(\mathbf{x}, \mathbf{y}), \quad (2)$$

where N_s is the number of scales and ncc_k operates on the k -th scale of the input image I . The similarity ζ results from multiple scale levels that have been taken at the closest position to the original position in the unscaled picture and the window size is kept constant, as shown in Fig. 5. In our empirical tests, we determined that a good trade-off between speed and robustness is a size of 15×15 pixels on 3 pyramid levels. This is not completely equivalent to the multi-sized similarity operator on the original image because it introduces implicit low-pass filtering by down sampling. Even though it is very robust while being relatively fast compared to using large similarity windows on the original image. In order to speed up the computation, an *early break* stops the evaluation of all pyramid layers if the similarity value on the higher pyramid level is below a certain threshold, since in practice most of the compared regions are not at all similar.

Finally, it is practically independent of the size of the input images and the size of the patterns. By using a constant window size the



Figure 5: Multiresolution similarity measure.

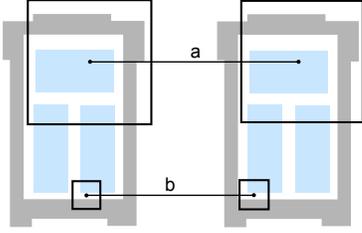


Figure 6: Two differently sized similarity windows with highly similar matches. a) A correct match with a window size of similar dimension with respect to the size of the sampled features. b) An example of a wrong match with a high similarity value caused by a too small sampling window.

multi-resolution similarity operator on the image pyramid is highly efficient compared to using large similarity windows on the original image.

Monte Carlo sampling. A common approach to dealing with complex or high-dimensional search spaces are Monte Carlo (MC) solutions. Using MC sampling to obtain samples of the data allows for a low-cost approximation of the expensive deterministic computation. Instead of computing the similarity for every pair of different locations, the Monte Carlo algorithm takes a statistical probe of the similarity at a number of random positions.

Facade elements such as windows, balconies, etc., are characterized by sharp orthogonal edges and corners. Based on this information we implement an importance sampling strategy. It is not so important to sample image regions without any salient features because they might not contain any facade elements. Instead we focus on edges and corners which are better indicators of facade elements. The implementation of such an edge-based importance sampling strategy is quite straightforward: an edge image is computed using Sobel-filtering and Canny edge detection [Canny 1986]. Using this sampling strategy, the accuracy of the result is significantly higher than for simple uniformly distributed random position sampling of the image, while requiring significantly less samples.

Distinguishing important patterns. We propose a sampling process to identify large image patterns, which casts a number of random samples and sorts the resulting offset into histogram bins if they meet certain criteria. The resulting histogram represents the distribution of similar offsets in the image. In order to identify these patterns and measure their offsets, we propose two different criteria to judge what is the best matching corresponding region for a given location: (1) the *threshold criterion* and (2) the *best match criterion*. In the following we introduce both criteria in form of their histogram classification functions $h(\Delta)$ and point out the respective pros and cons.

The *threshold criterion* simply defines a global threshold for the accepted similarity values. The histogram classification function $h(\Delta)$ with threshold criterion for N random samples and threshold t is given by:

$$h(\Delta) = \sum_i^N \begin{cases} 1 & \text{if } \zeta(p_i, p_\Delta) > t \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

This function counts how many samples (random pairs of points) with a given offset Δ have a multi-resolution similarity value greater than a fixed threshold t . We have determined empirically that the

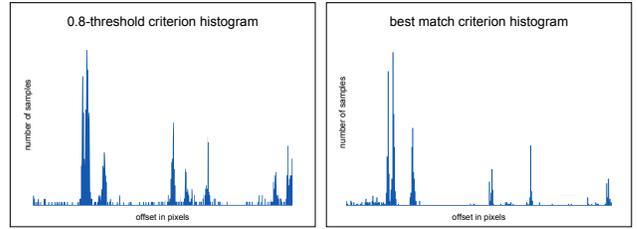


Figure 7: Comparison of histograms resulting from 100k samples with *threshold criterion* selection (left) and 1k samples with *best match criterion* selection (right). The broad peaks in the left hand histogram and high peaks of irrelevant offset combinations are signs of the much higher overall error of the simple *threshold criterion*.

threshold of 0.8 of normalized cross-correlation operator ensures that only highly similar matches are counted. By counting only samples with very high similarity values the variance of the estimated distribution of offsets is significantly lower. However, a quality criterion with a single fixed threshold still counts many imprecise matches because the sampled offsets are not compared to each other in any way. Even significant deviations from the perfect match of two regions may feature insignificantly high similarity values which might be much higher than the threshold. The problem arising from this fact is, that the results are noisy and the significant offsets may be hard to distinguish from the rest (see Fig. 7).

A more accurate criterion for finding the best recurrence of a spot in the image is the *best match criterion*. It compares the similarity values of multiple possible candidate offsets and chooses the best match. The idea is to draw more than one sample from one random location, compare them against each other and record only the best match which is the sample with the highest similarity value.

A definition of the histogram classification function $h(\Delta)$ implementing the best match criterion for N random samples from a uniform distribution is given as:

$$h(\Delta) = \sum_i^N \begin{cases} 1 & \text{if } \Delta = \arg \max_{\Delta_j} \zeta(p_i, p_{\Delta_j}), \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where all $\Delta_j \in \{D\}$. The range $\{D\}$ defines a set of all possible offsets in the current row or column of the image with respect to the current sample position.

To sample according to the *best match criterion* means to count how many times a given offset Δ_j is the best one in such that its multi resolution-similarity is higher compared to the similarity of any other offset at the sample location p_i . An offset with a high number of hits represents a pattern that is more dominant in terms of recurrence similarity and was found on a large image area.

Extraction of the relevant patterns. Typically, the dominant patterns are represented by a number of very similar offsets forming peaks in the histogram. These peaks are superimposed with random noise that might corrupt the results unless an appropriate evaluation method is used. To reduce the impact of noise, the histogram curve can be smoothed with a blur operator (i.e. a Gaussian kernel).

In this context it is also important to mention the optimal size of the filter kernel. While for small images up to one megapixels a 3-pixel filter kernel is sufficient it is certainly not adequate for a 10 megapixel image because it can no longer remove the large-scale noise. An optimal filter kernel size must therefore be derived from the size of the input image in order to adapt the filter kernel to the optimal relative size. In the reference implementation a filter kernel

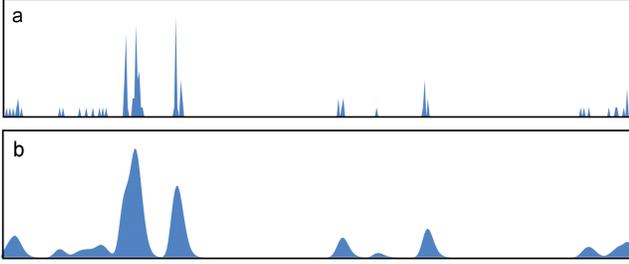


Figure 8: Original histogram (a) and a smoothed and normalized histogram (b). In the smoothed histogram some close peaks are merged together because of oversmoothing. This reduces the number of concurring extracted peak locations on the one hand but also degrades precision of the segmentation on the other hand.

size of $n = \frac{d}{50}$ proved to be useful for most images, where d is the current image dimension (width or height), depending of the processing direction. Finally, the peaks are obtained by *mean shift* clustering [Comaniciu and Meer 2002].

Post processing of extracted offsets. In many cases the extracted offsets include doubles, triples and higher multiples of the smallest offset to the first recurrence. If a pattern is not uniformly spaced throughout the image, which means that there are differently sized intervals between the re-occurring regions, it might as well happen that the extracted offsets contain combinations of those different offsets (see the annotations in Figure 9 for examples of multiples in a facade image). A simple but efficient solution to this problem is to remove all offsets that are close to integer multiples of the smallest offsets.

3.2 Localization and segmentation

We now know which patterns (given by their representative offset) are the prevailing ones in the image. Now we want to determine the location of each distinct repetitive pattern and its extent in the image.

The Similarity Curve. We again resort to an estimation using random sampling. The same multi-resolution similarity measure as used in the identification step serves as the criterion for the relevance of a specific pattern in a specific region. For every different offset the sampled data can be seen as a *similarity curve* containing the similarity values for every pixel row y or pixel column x in the image.

A horizontal similarity curve $S(x, \Delta)$ for an offset Δ is defined as follows: the image is sampled at every pixel column x at N random locations y_i . The mean over every pixel row is the value of the similarity curve at pixel column x (see Figure 10):

$$S(x, \Delta) = \frac{1}{N} \sum_i \zeta(p(x, y_i), p_\Delta(x, y_i)). \quad (5)$$

The definition of the *vertical* similarity curve is analogous to the horizontal curve in that for every image row y N samples x_i are drawn.

The localization of the patterns is done by comparing the similarity curves for each relevant offset against each other (see Fig. 10 top). By setting the curves in relation to each other, a decision can be made which image regions “belongs” to which pattern. Moreover,



Figure 9: Demonstration of a number of possible multiples of offsets A and B which might obscure the results of the histogram extraction. Two, three and four times multiples of an offset happen quite often and can be easily removed by postprocessing.

regions with very low similarity response to all major offsets are considered to be non-repetitive image regions.

Segmentation. The segmentation algorithm iteratively decides what is the most dominant offset in the local image region and then divides the image accordingly. The decision criterion for finding the most dominant offset of the next region is the accumulative similarity. In other words, the segmentation algorithm integrates over the similarity curve of every offset from the current position to the offset. This means that we need to integrate over a different interval for every offset. In order to be able to compare these accumulated similarity values against each other they need to be normalized by the offset. The offset with the highest normalized accumulated similarity wins and the size of the hereby segmented region is the offset. The current position advances to the end of this region and the algorithm enters the next iteration.

The iterative segmentation is defined formally by the position of the next splitting line L_{i+1} based on the position of the current splitting line L_i :

$$L_{i+1} = L_i + \arg \max_{\Delta} \left(\frac{\sum_{x=L_i}^{L_i+\Delta_j} S(x, \Delta_j)}{\Delta_j} \right), \quad (6)$$

where Δ_j are the relevant offsets that have been extracted from the image. L_0 is initialized to 0 or to the first row or column that exhibits significant repetitive response on any of the relevant similarity curves.

The highest value of the integral over the offset’s similarity curve normalized by dividing through the offset is used to decide at which offset to set the next splitting line, so to say, which offset represents the following region’s most dominant repetitive pattern best. As this method cannot account for intervals of non-repetitive nature it is necessary to identify the image regions where any of the offset’s similarity curve is below a certain threshold (i.e., 0.3) and apply the iterative segmentation algorithm to the remaining repetitive regions.

A shortcoming of this segmentation method is the fact that an offset Δ and its non-fractional multiple $N\Delta$, with $N = 2, 3, 4, \dots$, are treated as if they would represent completely different patterns, even if both offsets are occurring due to instances of a single pattern. This results in systematic errors when offsets are fighting with their multiples. Their similarity is quite equal yielding unstable results depending on the random numbers used for sampling. A possible solution is to modify the splitting function in order to slightly prioritize smaller offsets over larger ones with a weighting factor:

$$\omega(\Delta_j) = 1 - \left(\frac{\Delta_j}{\min \Delta} \varepsilon \right), \quad (7)$$

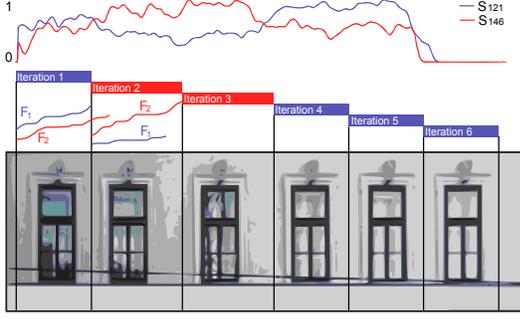


Figure 10: Illustration of the iterative segmentation algorithm. For each iteration and each major offset an integral F_i of the similarity curve S_j is calculated. Since the integration is over a different range for every offset, the resulting areas are normalized to allow a comparison. The offset with the higher normalized area wins the voting for this iteration. In this example in the first iteration the offset 121 is chosen, in the second iteration the offset 146 is selected, and so on.

where ϵ is a small penalty factor such as 0.2. Then the iterative segmentation function is given by:

$$L_{i+1} = L_i + \arg \max_{\Delta} \left(\frac{\sum_{x=L_i}^{L_i+\Delta_j} S(x, \Delta_j)}{\Delta_j} \omega(\Delta_j) \right). \quad (8)$$

The weighting function ω prioritizes the smaller offsets and hence effectively rules out unwanted multiples if their singular offset is present with a high similarity value. On the other hand, in case that an offset is the multiple of a smaller offset by accident but the local image area does not exhibit any smaller pattern then the larger one would still have a higher similarity value.

4 Performance

Best-match vs. Threshold criterion. Table 11 summarizes horizontal segmentation performance of a facade image with different resolutions using threshold sampling criterion with a threshold of 0.8 and 50.000 samples. The performance of vertical segmentation is equivalent to horizontal segmentation.

All timings presented here were recorded on a Intel Dual Core 2.4 GHz computer. The performance comparison shows the linear complexity of best-match sampling vs. the constant complexity of threshold sampling with respect to image resolution. It suggests that the best match criterion is best to be applied for small images while the threshold criterion is best suited for large images due to its constant complexity. On the other hand, the results of best-match criterion are more precise, so best-match sampling is better if high

Performance comparison, time (s)		
megapixel	best match	threshold
0,59	1,53	4,32
1,19	3,41	6,33
2,37	8,49	8,33
4,75	18,15	9,23
9,50	37,39	9,61

Table 1: Running time comparison.

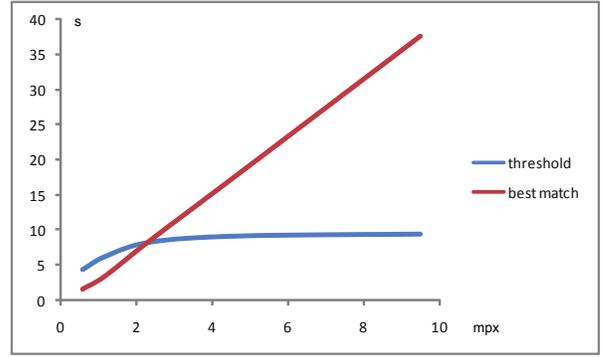


Figure 11: Performance comparison of the sampling criteria "best match" versus "threshold". The graph displays the running time of each sampling strategy as a function of image size.

precision is required, i.e., for images where the distance of different patterns which should be distinguished is relatively low.

Complexity. For best match sampling the complexity of the method depends on the number of samples n and the resolution of the image m in pixels. The algorithmic complexity for best match sampling is therefor limited by an upper bound of $O(nm)$ while the complexity of the threshold criterion depends solely on the number of samples taken. The size of the input image does not significantly influence the performance of the threshold criterion method. The algorithmic complexity for sampling with threshold criterion is therefore limited by an upper bound of $O(n)$, where n is the number of samples taken. If the number of samples is considered to be a fixed constant (because the number of samples does not dynamically change once an appropriate number has been chosen), then the complexity of "best match" is actually linear $O(n)$ with respect to image size n and the complexity of the threshold criterion is constant $O(1)$ for increasingly larger images.

Impact of the probe size. The performance of this segmentation method is not only dependent on the image size but also on the number of samples taken. Table 2 shows the horizontal segmentation performance of a typical facade image with a resolution of 0.4 megapixels and different numbers of samples. For the threshold sampling criterion, a threshold of 0.8 was used.

Parallelization. The algorithm is parallelizable in several ways to leverage of the computational power of contemporary multi-core processor architectures. For instance, one could divide the workload of the sampling stage by the number of processors available p , so that every thread takes $\frac{N}{p}$ samples individually in order to get a complete number of N samples. This approach does not require any

Best match criterion			Threshold criterion		
samples	time (s)	correct	samples	time (s)	correct
2	0,07	no	50	0,008	no
5	0,2	no	500	0,07	no
10	0,57	yes	1.000	0,12	no
20	0,81	yes	2.000	0,25	yes
40	1,64	yes	5.000	0,71	yes
60	2,19	yes	10.000	1,29	yes
80	3,15	yes	20.000	2,63	yes
100	3,99	yes	50.000	6,59	yes
200	7,01	yes	100.000	12,91	yes
500	18,87	yes			
1000	36,67	yes			

Table 2: Probe size dependence.

synchronization between the independent processing threads until the end when the histogram is evaluated. The individual histograms of each thread can be merged for the extraction of the major offsets.

5 Quality

The precision of the segmentation method presented in this paper is given by the average deviation from the exact solution on an appropriate number of test cases. For this purpose the algorithm has been tested against a hand-crafted image with exactly spaced instances of a pattern. The following table lists the average deviation of 50 runs each for both sampling criteria as a percentage of the exact solution.

The slight fuzziness of the segmentation results are due to the applied Monte Carlo random sampling. For example, if the windows on a facade image are spaced by an offset of 300 pixels, then a 2% deviation means that the resulting detected offsets may be off by 5 pixels. The relative representation of the error as percent of the exact result has been chosen because the absolute error grows proportionally with the absolute size of the patterns.

Resolution independence. The current implementation is able to successfully segment facade images starting from a lower limit resolution of 100 kilopixels up to extremely large images which are bound only by the memory capacity of the machine. Due to the adaptive multi-resolution sampling the segmentation results are very stable for an image under extremely different resolutions.

All parameters are defined relative to image dimensions. The advantage of such an approach is that the algorithm automatically adapts to the resolution of the input image and yields correct results without tweaking any parameters.

Of course, results are always more precise on high-resolution images. It may happen, that on low-resolution images not all offsets are measured correctly because they are either smaller than the smallest correlation window in the image pyramid or they are too close to other offsets and their peaks are merged during histogram smoothing. For good results a minimum resolution of one megapixel is suggested for use of this method, although in certain cases it has been observed to work quite well with much lower resolution images.

Robustness to Gaussian blur. The robustness with respect to typical image damage is demonstrated by showing the results of tests against incrementally more blurry and noisy versions of the same picture. The following table compares the robustness to blurriness of the best match sampling method with the threshold method.

Under extreme blurring the importance sampling strategy fails and too few samples are drawn. This is due to the method's focus on image discontinuities such as edges and corners. With increasing blur such image features vanish. Nevertheless, the method can be considered robust against blurriness.

Robustness to random noise. The following table compares the robustness of the best-match sampling method against the threshold method with respect to overlaid random noise.

	best-match	threshold
average error	1.67%	1.66%
standard deviation	0%	0.35%

Table 3: Precision. See description in the text.

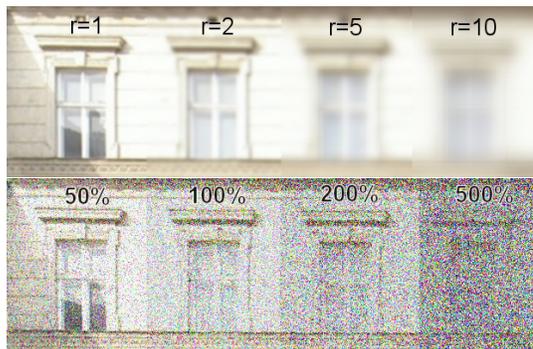


Figure 12: The test images: under Gaussian blur with different radii (top) and under increasing levels of random noise (bottom).

Obviously the two different sampling methods behave completely different with random noise applied to the input images. The best-match sampling criterion is extremely robust and is even under heavy interference with random noise able to find the regular pattern beneath. Threshold sampling, on the other hand, is quite fragile with noisy images. This is due to the fixed similarity threshold criterion, which must be fulfilled for each sample in order to be stored in the histogram. In order to perform well with degrading image quality and noise, this threshold would need to be adapted dynamically. This would be a possible subject of further improvement. Figure 13, bottom, demonstrates the robustness of the segmentation algorithm on a real-world image – the algorithm reliably detects the repetitive pattern even though it is heavily obscured by blur and irregular vegetation.

6 Conclusions

We proposed a novel method for fast recognition of repetitive patterns along horizontal and vertical axes of the image. The method is entirely based on the assumption that explicit analysis of the image content could never lead to a generalized method and that measurement of repetitive similarities is enough to identify and segment facade elements. As the results show, this approach was successful, both in a reliable and efficient manner. However, by using only information on the translational symmetry of a set of random image locations it is not possible to discriminate certain areas as background signal and identify others as foreground. In other words, by not analyzing the content we are not able to identify any concrete objects in the image or distinguish them from uninteresting background noise.

For future work we see room for speed improvements of the Monte Carlo sampler by applying more sophisticated importance sampling of the underlying PDF. An additional possible extension is the introduction of a finer similarity measure for windows based on local reflective symmetry, which is extensively present on typical facades.

radius	best-match correct	threshold correct
1	yes	yes
2	yes	yes
5	yes	no
10	no	no

Table 4: Robustness to Gaussian blur.

noise (%)	best-match correct	threshold correct
50	yes	yes
100	yes	no
200	yes	no
400	yes	no
600	no	no

Table 5: Robustness to random noise.

Acknowledgments

We would like to acknowledge financial support from WWTF grant number C106025 as well as from NSF and NGA. Finally, we would like to acknowledge the Aardvark-Team in Vienna.

References

- ALEGRE, F., AND DELLAERT, F. 2004. A Probabilistic Approach to the Semantic Interpretation of Building Facades. In *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres, 2004*.
- ALI, H., SEIFERT, C., JINDAL, N., PALETTA, L., AND PAAR, G. 2007. Window detection in facades. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, 837–842.
- BAILEY, D. G. 1997. Detecting regular patterns using frequency domain self-filtering. *IEEE Computer Society*.
- BOIMAN, O., AND IRANI, M. 2007. Detecting Irregularities in Images and in Video. *International Journal of Computer Vision* 74, 1 (Jan.), 17–31.
- BRENNER, C., AND RIPPERDA, N. 2006. Extraction of Facades using rjMCMC and Constraint Equations. In *PCV '06, Photogrammetric Computer Vision*, ISPRS Comm. III Symposium, IAPRS, Bonn, 155–160.
- CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (November), 679–698.
- CECH, J., AND SARA, R. 2008. Windowpane Detection based on Maximum A Posteriori Probability Labeling. In *Image Analysis - From Theory to Applications*.
- COMANICIU, D., AND MEER, P. 2002. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, 5, 603–619.
- DICK, A., TORR, P. H. S., AND CIPOLLA, R. 2004. Modelling and Interpretation of Architecture from Several Images. *International Journal of Computer Vision* 60, 2 (Nov.), 111–134.
- HAN, F., AND ZHU, S.-C. 2009. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 1 (Jan.), 59–73.
- HSU, J. T., LIU, L.-C., AND LI, C. 2001. Determination of structure component in image texture using wavelet analysis. In *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, IEEE, 166–169.
- KORAH, T., AND RASMUSSEN, C. 2007. 2D Lattice Extraction from Structured Environments. In *2007 IEEE International Conference on Image Processing*, IEEE, vol. 2, II – 61–II – 64.
- LEE, S. C., AND NEVATIA, R. 2004. Extraction and integration of window in a 3D building model from ground view images. In *Proceedings of the 2004 IEEE Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 2, 113–120.
- LIU, Y., COLLINS, R. T., AND TSIN, Y. 2004. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 3 (Mar.), 354–71.
- LOY, G., AND EKLUNDH, J.-O. 2006. Detecting Symmetry and Symmetric Constellations of Features. In *Computer Vision - ECCV 2006*, Springer Berlin Heidelberg, vol. 3952, 508–521.
- MAYER, H., AND REZNIK, S. 2006. MCMC Linked with Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences. In *PCV '06, Photogrammetric Computer Vision*, ISPRS Comm. III Symposium, IAPRS, 130–135.
- MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics* 25, 3 (July), 560.
- MÜLLER, P., ZENG, G., WONKA, P., AND VAN GOOL, L. 2007. Image-based procedural modeling of facades. *ACM Transactions on Graphics* 26, 3 (July), 85.
- MUSIALSKI, P., WONKA, P., RECHEIS, M., MAIERHOFER, S., AND PURGATHOFER, W. 2009. Symmetry-based facade repair. In *Vision, Modeling, and Visualization Workshop 2009 (VMV 2009)*.
- PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. J. 2008. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics* 27, 3 (Aug.), 1.
- SCHINDLER, K., AND BAUER, J. 2003. A model-based method for building reconstruction. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003.*, IEEE Comput. Soc, 74–82.
- SHECHTMAN, E., AND IRANI, M. 2007. Matching Local Self-Similarities across Images and Videos. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Minneapolis, MN, USA, 1–8.
- TSAI, F., LIU, J.-K., AND HSIAO, K.-H. 2006. Morphological Processing of Video for 3D Building Model Visualization. In *Proceedings of 27th Asian Conference on Remote Sensing (ACRS2006)*, Ulaanbaatar, Mongolia, 1–6.
- TURINA, A., TUYTELAARS, T., MOONS, T., AND VAN GOOL, L. 2001. Grouping via the Matching of Repeated Patterns. Springer, S. Singh, N. Murshed, and W. Kropatsch, Eds., *Lecture Notes in Computer Science*, 250–259.
- VAN GOOL, L., ZENG, G., VAN DEN BORRE, F., AND MÜLLER, P. 2007. Invited paper: Towards mass-produced building models. In *PIA07*, 209.
- WANG, X., AND HANSON, A. R. 2001. Surface Texture and Microstructure Extraction from Multiple Aerial Images. *Computer Vision and Image Understanding* 83, 1 (July), 1–37.
- XIAO, J., FANG, T., TAN, P., ZHAO, P., OFEK, E., AND QUAN, L. 2008. Image-based façade modeling. *ACM Trans. Graph.* 27, 5, 1–10.
- XIAO, J., FANG, T., ZHAO, P., LHUILLIER, M., AND QUAN, L. 2009. Image-based street-side city modeling. *ACM Trans. Graph.* 28, 5, 1–12.



Figure 13: Results. The red lines indicate the grid that has been automatically detected on each facade (best seen in color). Note, images on the bottom demonstrate the robustness of the algorithm on a facade that is obscured by trees and a facade with different reflections in the windows.