Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

FAKULTÄT FÜR !NFORMATIK

Automatic Recognition of Repeating Patterns in Rectified Facade Images

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computergraphik/ Digitale Bildverarbeitung

eingereicht von

Meinrad Recheis
Matrikelnummer 0026003

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung:
Betreuer: Prof. Werner Purgathofer
Mitwirkung: Dipl.-Mediensys.wiss. Przemyslaw Musialski

Wien, 16.12.2009

(Unterschrift Verfasser) (Unterschrift Betreuer)

Meinrad Recheis, Seitenberggasse 29/2, 1160 Wien

"Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe."

Wien, 16.12.2009, ____________________________

Meinrad Recheis

Automatic Recognition of Repeating Patterns in Rectified Facade Images

16. Dezember 2009

Acknowledgments

I would like to express my gratefulness to all who contributed to this work in some way. Especially, I would like to thank my supervisor Przem Musialski for his advise, the many inspiring conversations and his patience with me.

Many thanks to Robert Tobler and Alexander Wilkie whose incredible lectures on advanced rendering techniques taught me some sophisticated concepts I could successfully apply during this thesis.

A special loving thank you to my dear wife who encouraged me to go on especially when I seemed to be at a loss.

Last but not least, a warm Thank You to my parents. They did everything to support my education which made this work possible in the first place.

Danksagung

Ich möchte mich bei Allen ganz herzlich bedanken, die in irgendeiner Form zu dieser Arbeit beigetragen haben. Ganz besonders möchte ich meinem Betreuer Przem Musialski für seine Tips, die vielen inspirierenden Gespräche und seine Ruhe und Geduld danken.

Vielen herzlichen Dank an Robert Tobler und Alexander Wilkie deren unglaublich interessante Vorlesungen über spezielle Renderingtechniken ich genießen durfte. Durch sie hatte ich Zugang zu einigen interessanten Konzepten die ich bei dieser Arbeit erfolgreich einsetzen konnte.

Ein ganz liebevolles Dankeschön an meine Frau, die mir wertvollen moralischen Beistand geleistet hat, wann immer ich nicht mehr weiter wusste.

Zum Abschluß möchte ich noch meinen Eltern ganz herzlichen Dank aussprechen. Ihr unermüdliches Bestreben mich zu fördern, hat diese Arbeit im Grunde erst möglich gemacht.

Abstract

Building facades typically consist of multiple similar tiles which are arranged quite strictly in grid-like structures. The proposed method takes advantage of translational symmetries and is able to analyze and segment facades into tiles assuming that there are horizontal and vertical repetitions of similar tiles. In order to solve this quite complex computer vision task efficiently a Monte Carlo approach is presented which samples only selected image features. This method, which is meant to be a preprocessing step for more sophisticated tile segmentation and window identification in urban reconstruction tasks, is able to robustly identify orthogonal repetitive patterns on rectified facade images even if they are partially occluded, shadowed, blurry or otherwise damaged. Additionally, the algorithm is very running time efficient because neither quality of results nor the computational complexity are significantly depending on the image size.

Zusammenfassung

Gebäudefassaden bestehen üblicherweise aus einer regelmäßigen gitterförmigen Anordnung von ähnlichen Elementen. Diese Eigenschaft, nämlich die Translationssymmetrie von Fenstern auf Fassaden, wird von der vorgeschlagenen Segmentierungsmethode ausgenutzt um ohne aufwändige Analyse des Bildinhalts Rückschlüsse auf die Anordnung der Fenster zu ziehen und diese anschließend in ähnliche Stücke aufzuteilen, sofern sie sich horizontal oder vertikal wiederholen. Um diese relativ komplexe Aufgabenstellung der Computer Vision effizient umzusetzen wird ein Monte-Carlo-Ansatz präsentiert, welcher aus einer Menge von speziell ausgewählten Features im Bild zufällige Stichproben nimmt. Die präsentierte Segmentationsmethode, die als Vorverarbeitungsschritt für andere Algorithmen zur Erkennung von Fassadengeometrie für Stadt-Rekonstruktionsprozesse dienen soll, ist äußerst robust bei der Identifikation von sich wiederholenden Bildmustern in rektifizierten Fassadenaufnahmen, selbst wenn diese von Gegenständen verdeckt, beschattet, verschwommen oder anders beeinträchtigt sind. Außerdem sind die Algorithmen äußerst Laufzeiteffizient ausgestaltet, da weder die Qualität der Ergebnisse noch die Rechenkomplexität wesentlich von der Bildgröße beeinflusst werden.

Contents

Chapter 1.

Introduction

1.1. Motivation

Automatic urban reconstruction is currently under intensive research in the Computer Vision community. One of the still challenging tasks is the recognition and reconstruction of facade-details like windows and balconies. These are considered key elements of realistic representations of building facades. The appearance of reconstructed building geometry texture mapped from photographs gains a lot of realism from the displacement of windows, balconies and other coarse geometric detail. Otherwise the reconstructed facade looks strangely flat when viewed from certain angles especially in walk-through scenarios.

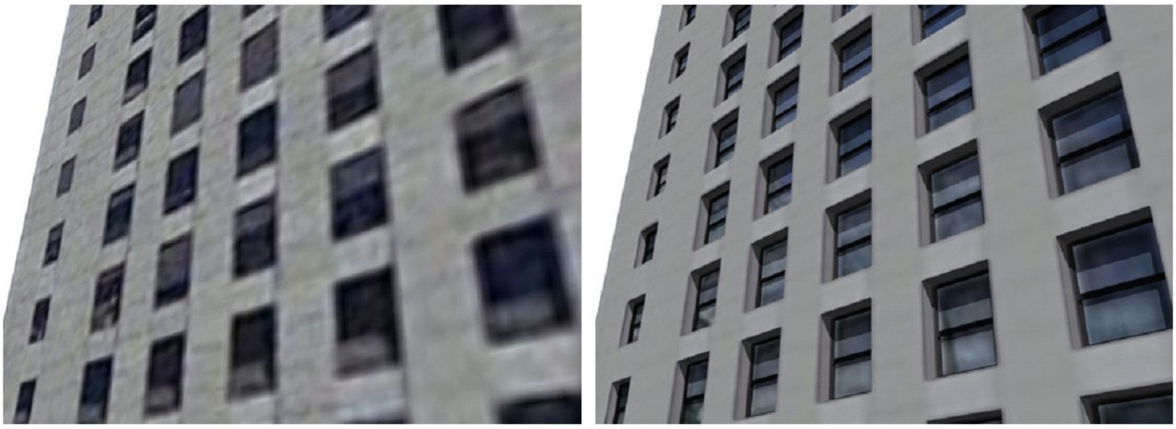


Figure 1.1.: Left: Non-realistic impression of a texture mapped building model when viewed from a grazing angle nearby the building. Right: Highly increased realism by modeling coarse geometric detail of the facade which casts shadows (and using high-resolution textures). Image courtesy of [MZWVG07].

In the last ten years a variety of approaches to urban reconstruction and facade element recognition have been developed. Raw building geometry is acquired either by terrestrial laser range finders or reconstructed from satellite, airborne or ground based images or video. Facade details like windows are mostly extracted from images and then superimposed on the coarse building models. However, fully automatic methods are still utopia. The vast number of different architectural forms and styles makes it extremely difficult to develop a method that works for every urban environment without requiring a fair amount of user input.

The VRVis Company in Vienna is actively researching novel methods for computer aided Urban reconstruction based on ground based photographs. VRVis is participating and contributing major parts to the Wiki-Vienna project, which intends to create a platform for reconstruction and visualization of the city of Vienna. In this context, this thesis explores a new approach for automated content-independent segmentation of building facades into repetitive tiles, where each tile contains just one single feature. The presented method is designed to be a preprocessing step which serves as the base for a sophisticated tile segmentation and facade geometry reconstruction algorithm.

1.2. Related Work

The problem of facade image analysis for reconstruction purposes has been tackled by many researchers in the last ten years. Many different approaches for extraction of facade structure, facade elements and facade geometry have been proposed. This section structures the literature of related work in order to give a good overview of all the different techniques that have been applied or created to solve problems in the class of recovering facade details, depth information and compositional structure.

In the literature, there is a clear trend towards mass-produced city models and there is also a trend towards approaches that require less redundant and easy obtainable input data like photos. A good example for this trend towards simplification of input acquisition is [VG07], demonstrating a sophisticated method that requires only one single input image with strong enough perspecitve distortion. Another observation that can be made is that city reconstruction applications, as soon as they are developed to the point where they become usable products instead of research toys, are showing a greater demand of automation in order to cope with the vast amounts of data. The following discussion of the state of the art attempts to judge the degree of automation of the proposed solutions, which is not always easy because many authors conceal information regarding this matter.

In the next few sections, the problem solution strategies are categorized by their general approach and briefly explained. Most methods approach the facade reconstruction problem as an image segmentation problem, others define it as a feature detection challenge. Then again, some of the proposed solutions use both approaches in different steps of their processing pipelines. Finally, a subsection is devoted to the treatment of irregularities and damages in input data.

1.2.1. Reconstruction from Multiple Views versus Single View

Almost all of the earlier methods are based on multiple views of the same facade for several reasons. The most approaches infer a geometric 3D-model of a facade and its details using well known image matching and triangulation algorithms [WTT+02], [SB03], [DTC04], [MR07]. For example, Wang *et al.* [WTT+02] conduct a simple depth estimation for automatic displacement of the recognized windows from the facade plane. Schindler and Bauer [SB03] calculate a dense point cloud from multiple views. After fitting planes against the point clouds representing the facades they classify outliers which are displaced from the main planes as window regions.

Other uses of multiple images have been proposed by Wang *et al.* [WTT+02] and Tsai *et al.* [Tsa06] who merge overlapping building textures to handle occlusions, image noise or differences in shading. Furthermore, by searching for the same features in multiple views, image recognition tasks such as finding good window outlines appear to be significantly more robust [SB03], [DTC04].

Many of the multi-view methods yield fully automatic building reconstructions for specific classes of buildings provided that enough images from many different angels are given. On the contrary, it is very hard to find an automatic solution based on only a single image. Although, many of the more resent publications describe methods which require only a single input image. Some of them need user input or additional data to accomplish their tasks. For example, Brenner and Ripperda [BR06] use 3D data from a laser range scanner to classify certain image regions as windows due to their small depth displacement relative to the facade plane. Müller *et al.* [MZWVG07] developed a method that automatically infers a hierarchical grammar based model of the facade from which detailed 3D facade models can be generated. Still, they need user input to adjust the depth of specific facade elements like windows sills, balconies etc. Lee and Nevatia [LN04], on the contrary, have an automatic solution for estimating the depth of single facade elements from a single image. They solved it by conducting a plane sweeping search. On the other hand, their method is also not fully automatic because they need a 3D building model in order to rectify their facade images.

The one and only fully automatic single-view-based method has been published by Van Gool *et al.* [VG07]. They extend Müller's method for images with strong perspective, from which they are able to reconstruct the 3D geometry of many common facade details without the need for user interaction. Their method is restricted to images with sufficient perspective distortions though.

1.2.2. Segmentation Methods

The earlier works are based on morphological segmentation, for example [WTT^{+}02], which applies oriented region growing (ORG) to segment dark regions (usually windows) on light facades. The assumption that windows are darker than their surrounding facade is, however, weak and may work well only for airborne pictures. Ground based photographies often reflect buildings or the bright sky, especially when shot in an urban environment. Another use of morphological segmentation is presented by Tsai *et al.* [Tsa06] who calculate a greenness index (GI) to identify and suppress occlusions by vegetation on their facade textures which they extract from drive-by video recordings. On the cleaned textures they also apply ORG to find dark window regions.

A quite easy and robust segmentation of window panes is possible on dense point clouds resulting from image based triangulation or triangulations of laser range scanners [WTT^{+}02], [SB03], [BR06]. Since most windows are displaced behind the wall the majority of facades can be reconstructed. To improve the segmentation results and provide a simple facade description all solutions apply template matching of simple geometric facade features models against the segmented points. To model windows the majority of methods usually use rectangular shapes. Models for arches have also been proposed [SB03], [LN04].

Monte Carlo Marcov Chain approaches. The most promising class of solutions are hierarchical rule based segmentation algorithms. They cut down a facade into small irreducible parts which are arranged according to hierarchical context free grammar rules. Alegre and

Dellaert [AD04] first proposed a specific set of grammar rules and a Monte Carlo Marcov Chain (MCMC) approach to optimize the parameters in order to fit the hierarchical model against the facade image. Yet, the model they provide does not generalize to a large class of building facades. Brenner and Ripperda [BR06] show that it is possible to infer a hierarchical model from grammar rules and constraint equations using reversible jump Monte Carlo Marcov Chain (rjMCMC) for parameter optimization and rule tree inference. The approach seems promising but the solution is very complex, requires laser scan data of the facade and seems not yet good enough for mass production of building models. For facades that are asymmetric or based on complicated rules this method is the only feasible one. But the main class of facades is very simple and regular. A single-view approach for rule based segmentation of simple regular facades has been published by Müller *et al.* [MZWVG07]. They split the facade image into floors and the floors into tiles. the tiles are then split by some simple rules to approximate the window outlines. As splitting criterion they define an energy functional over the local density of vertical and horizontal edges. While this is a very robust method, it works only for the class of highly regular and symmetric facades that are based on such simplistic grammar rules. Annother MCMC method that should be mentioned here but is discussed in the next section is [DTC04].

1.2.3. Feature Detection Approaches

A good example for the texture-based feature detection approach is given by Dick *et al.* [DTC04][1]. They assume, for example, that image regions of windows have many strictly orthogonal texture elements or that Greek pillars have only strong vertical texture elements. Of course, this solution is only applicable for a very limited group of buildings. Similar and also very limited are window detection approaches that rely on the fact that sometimes windows are darker than their surrounding walls [WTT$^+$02], [Tsa06], [BR06].

Lee and Nevatia [LN04] published a window detection method that uses only edges. First they project the edges horizontally and vertically to get the marginal edge pixel distributions. They assume that these have peaks where windows are located. From the thresholded marginals they construct a grid which approximates the window outlines. They then match the window outlines against the image edges to detect the correct outlines of the windows. Later, they detect if their found windows happen to have arches and try to fit a general arch-model against the image evidence.

[1] This work is not easy to classify as it is also making use of Monte Carlo Marcov Chains.

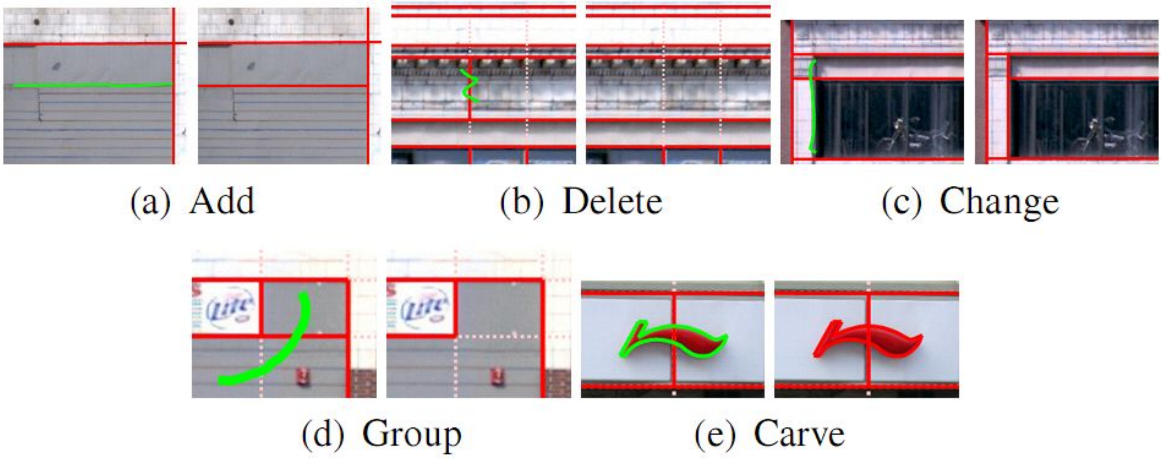
Many methods rely on template matching to model windows and other facade detail. Schindler and Bauer [SB03] match shape templates against point clouds. Mayer and Reznik [MR07] efficiently match template images from a manually constructed window image database against their facades. Müller *et al.* [MZWVG07] match appearance of their geometric 3D window models against facade tiles. The advantage of template matching is that reconstruction results look very realistic but, on the other hand, these reconstruction results are in most cases not authentic because there is no template data base that contains all possible shapes. Some have also combined template matching with machine learning, like Ali *et al.* [ASJ+07] who showed that it is possible to train a classifier such that it identifies a high percentage of windows even in images with perspective distortion.

Given the fact, that the majority of windows and other facade elements are rectangular, a common approach to facade reconstruction is searching for rectangles or assuming that all windows are rectangular. Almost all methods discussed here somehow assume rectangular shapes in some stages of their algorithms but do not solely rely on it. A recent approach that bases solely on rectangle detection is the window-pane detection algorithm by Cech and Sara [CS07] which identifies strictly axis-aligned rectangular pixel configurations. Their algorithm in fact solves a NP-hard optimization problem which is approximated sufficiently.

Finally, the repetitive nature of facade elements can be exploited to identify them. Van Gool *et al.* [VG07] search for similarity chains in perspective images to identify repeated facade elements.

1.2.4. Treatment of Irregularities

Quite a lot of the referred works assume, that facade structures are more or less regular. Some even use this widely shared property of facades to fix recognition errors or missing data, like Wang *et al.* [WTT+02] who introduced a periodic pattern fixing algorithm (PPF) or Tsai *et al.* [Tsa06] who replicate features along detected mirroring axes to fix occlusions by vegetation. The common approach to cope with unexpected irregularities in the data introduced by eccentric facade designs or missing data resulting from occlusions is to rely on certain final user interactions. The bottom line is, as time of this writing, there are no known city reconstruction methods which are really completely automatic for a wide variety of input data but some of them do quite a good job in reducing the necessary user interactions to a minimum.



(a) Add

(b) Delete

(c) Change

(d) Group

(e) Carve

Figure 1.2.: Geometric facade detail modeling interactions support difficult cases which the automatic process cannot handle correctly. Images courtesy of [XFT$^+$08].

1.2.5. General Approaches to Processing of Repetitive Patterns

Apart from these highly specialized methods for building reconstruction there are other more general works which are related to this thesis because they deal with detection of repetitive patterns in images. Bailey [Bai97] shows that it is possible to detect repetitive image patterns by self-filtering in the frequency domain. He is able to reconstruct missing data in highly repetitive images. This approach could be used to reconstruct missing data in occluded facades.

Turina *et al.* [TTVG01] detect repetitive patterns on planar surfaces under perspective skew using Hough transforms and application of various grouping strategies. They also demonstrate some good results on building facades but there is no application for urban reconstruction using this approach yet.

Han and Zhu [HZ05] 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. Results on facade images show, that rectangular windows are detected quite robustly due to the grammatical constraints. Further development for reconstruction purposes seems to be promising.

Boiman *et al.* [BI07] detect irregularities in images using cross-correlation and Shechtman and Irani [SI07] apply a similar approach to identify local self similarities in images from which they generate very robust feature descriptors. The two methods could be used to detect occlusions in facade images and to identify repetitive facade elements by shape template matching.

Hsu *et al.* [HLL01] use wavelet decomposition of the autocorrelation surface to segment a regular image into tiles. Liu *et al.* [LCT04] detect crystallographic groups on repetitive image patterns using a sophisticated dominant peak extraction method for extraction of maxima in the autocorrelation surface. They also achieve a segmentation into non-repetitive regions. The most advanced local repetitive pattern detection algorithm is given by Hays *et al.* [HLEL06] which is able to detect even distorted regular grids. These repetitive pattern segmentation algorithms, however, cannot be directly used for facade window detection because the patterns created by repetitions of windows and intermediate walls cannot be split at any possible location. For window segmentation a method is needed, that reliably splits between the windows, as shown in [VG07].

Chapter 2.

Automatic Recognition of Repeating Patterns in Rectified Facade Images

2.1. Introduction

The process of urban reconstruction comprises many stages such as data acquisition, reconstruction of building geometry, texture generation and efficient rendering, each of which are subject to active research in the community. Especially the topic of automatic reconstruction of structural facade details, such as windows, is one of the many open topics in the urban reconstruction field. It has received increasing attention of international research groups but because of the complex problem there are not yet any sufficiently efficient and at the same time highly generic and automatic solutions.

The contribution of this thesis addresses this topic and presents a novel approach for recognition and segmentation of repetitive facade details. Due to the fact that repetitive patterns are independent of appearance and architectural style of facade elements this segmentation method is shown to be outstandingly robust with respect to common unpredictable variations in input images such as shadows, cables, excessive decorative elements or variations of appearance caused by pollution.

Considering the high volumes of data that city reconstruction systems have to deal with it is particularly important for any of the processing steps to be highly optimized and fast. The introduced method is able to compute a facade tile segmentation very efficiently, even for high-resolution images. The presented method is designed to be a generic preprocessing step for specialized facade detail fitting and reconstruction algorithms.

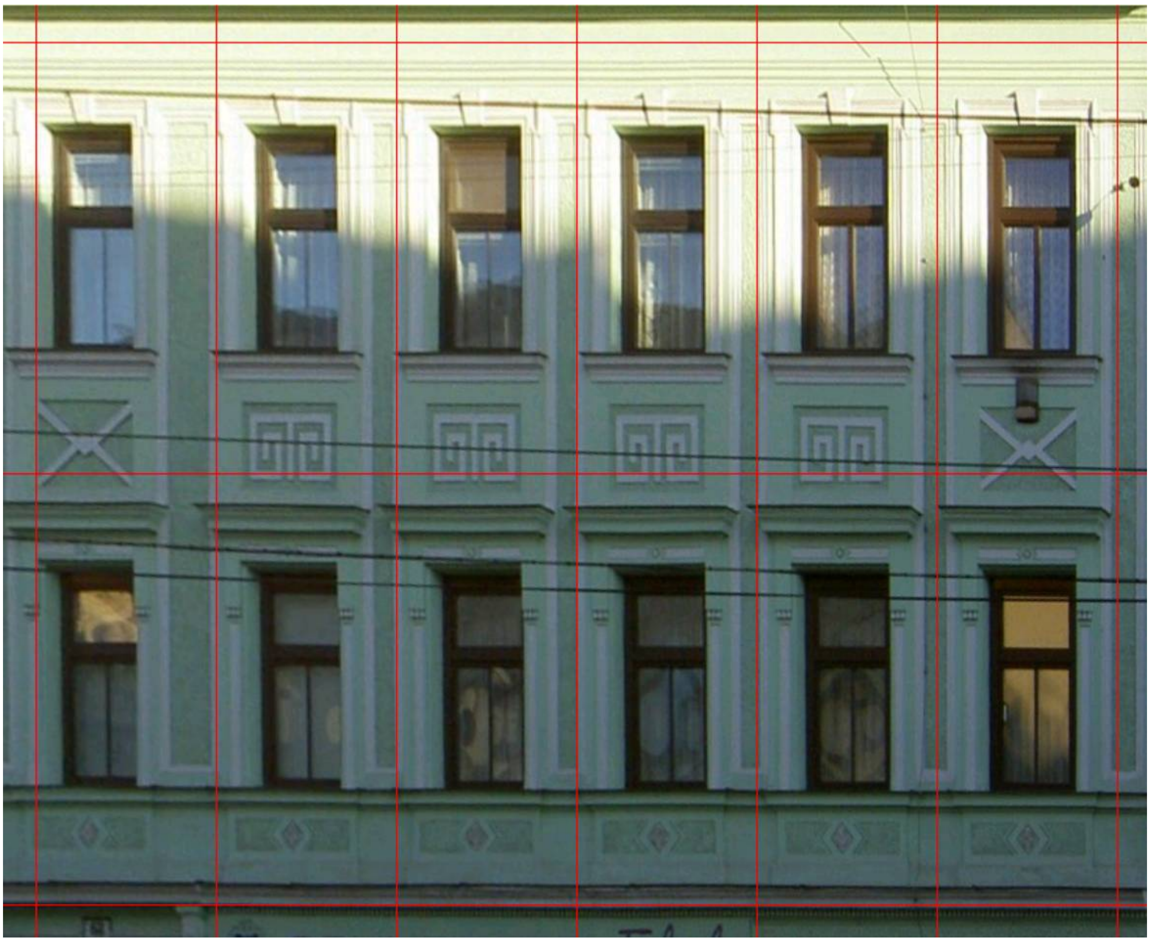


Figure 2.1.: Window tile segmentation on partially shadowed facade (detail). The computed segmentation is illustrated by red lines superimposed on the original image. The segmentation is not obscured by shadowing or other potentially disturbing image artifacts (see chapter 3).

The facade segmentation method presented here is a particular part of a large process for automatic urban environment reconstruction. To give the reader a rough overview of the whole process and a context for this work this process is shortly summarized.

Data acquisition. First, data about the city to be modeled has to be acquired. In most of the cases detailed CAD models are not available so that the complete geometry has to be reconstructed from laser scans or from multiple overlapping photographs. The latter data source is getting more popular lately because it shows great potential when opened for public contribution via the wiki paradigm. The high volumes of photos required for detailed reconstruction have to be handled efficiently giving rise to advanced image database concepts.

In the case of the Wiki-Vienna research project at the VRVIS the input images come from ground based photographs shot by people walking in the streets using standard consumer photo cameras. Today almost everyone is carrying a cell phone which is equipped with a camera.

The images are captured with cameras and are of course subject to lens distortion. The exact camera parameters are not necessarily known because it is not practicable to calibrate everyone's camera, especially if photos from the public are accepted. In this case, it is a great challenge to apply automatic inference of lens parameters in order to undistort the photographs.

Image matching and triangulation. For the multiple view reconstruction approach it is required to find pairs of overlapping images from the large pool of photos. Use of Geo tags might reduce the search complexity. Clever feature extraction and descriptor hashing techniques are being applied to reduce the potentially high computational complexity of this task. Some have also used successive frames from video instead of photographs to overcome the expensive image grouping step. The registration of overlapping photos against each other is typically done via comparison of SIFT features [Low03] in both images .

From such identified multiple overlapping views it is then possible to infer camera positions and triangulate 3D point clouds of the photographed buildings.

Coarse geometry reconstruction. Such point clouds are not yet presentable using a point based surface rendering technique because they are quite noisy due to the limited precision imposed by image resolution, number of matching image features and missing points caused by optical occlusions in the input images. Therefore, it is needed to find high-level geometric models for these point clouds which is another challenging task. The typical approach is to identify high-level geometric primitives such as planes and convert the point cloud into a lower resolution mesh of building blocks.

Visualizations of the resulting coarse building block models textured with projections of the original photos yet lack a crucial property that is necessary for a realistic impression: fine grained geometric facade details such as window sills which can be modeled as displacements from the facade planes. The sun usually casts shadows from window sills on the walls and the glass windows usually reflect other buildings or the sky. If these effects are missing visualizations of urban environments are perceived quite unrealistic.

Detail geometry reconstruction. The identification and reconstruction of such facade details can only be conveniently done in the original input photos. For this purpose the photos are projected onto the identified facade planes effectively reverting the perspective distortion. This process is called rectification or orthogonalization [TS08]. Such orthogonalized facade images are much easier to process than the original photos which are perspectively distorted. See figure 2.2 for an example.

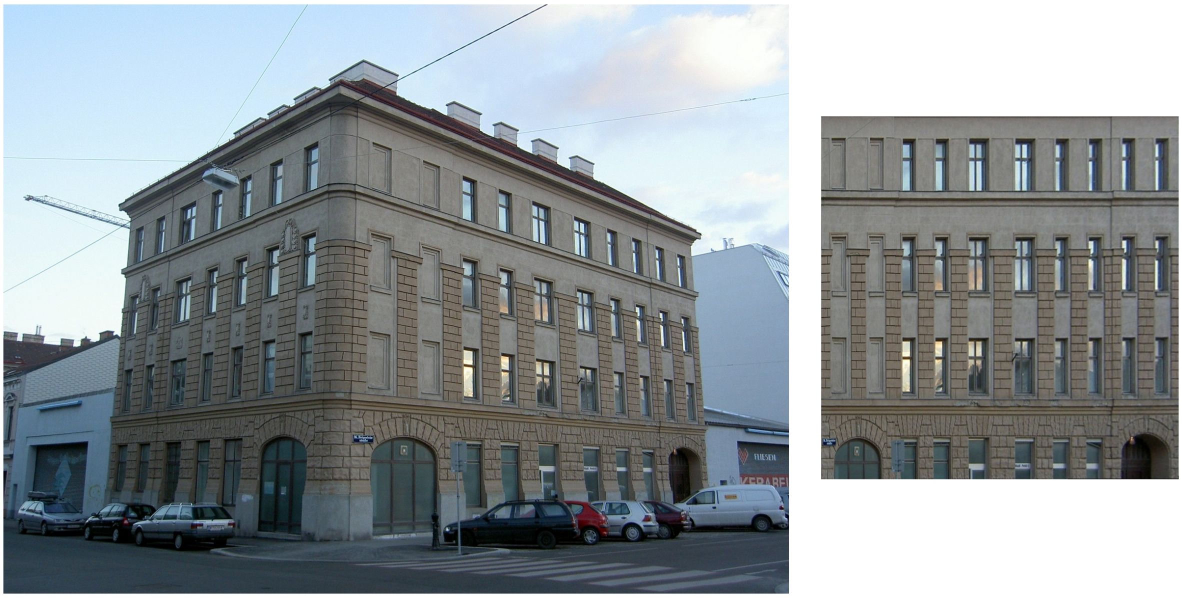


Figure 2.2.: Original photo and extracted rectified facade. This facade image has been extracted by manually by specifying four corresponding points in both images and warping the image using a linear transformation as described in Appendix A.

This is the point where the fast facade segmentation method presented in this thesis may be applied. The output tiles can then be further segmented to reconstruct size and location of facade details like windows, balconies, doors and the like. The geometry of the identified facade details can be modeled (i.e. by choosing from a database of 3D CAD-models of windows. Also, a procedural grammar might be derived which efficiently describes the repetitive alignment of facade tiles and reduces the memory footprint of the model.

Rendering. The reconstructed 3D geometry of a city is so enormously large that highly sophisticated rendering methods are required to allow for interactive walk-through scenarios or other real time visualization applications. Even high-end graphics hardware can not handle models of such scale. To prevent the graphics card memory from overflowing sophisticated visibility culling strategies dynamically render only the visible parts of the city model with respect to the viewer's location. Also very important are work load reducing automatic level-of-detail algorithms which reduce the geometric resolution of visible elements according to the screen-space they currently occupy. For instance a highly detailed building model that is projected onto an area as small as a few pixels can be temporarily reduced to a simple cube or silhouette-polygon without loss of perceived visual accuracy.

This work presents a contribution to the sub-process of recognition of geometric details in orthogonalized facade images. The next sections describe the general idea and give a brief overview of the approach.

2.1.1. Idea

The main idea behind the proposed method is to exploit the inherently repetitive nature of almost all facade elements to identify facade tiles, locate them and finally partition the facade image into tiles. The approach to only use similarity as segmentation criterion was driven by the challenge to segment typical art-nouveau facades of the inner city in Vienna. Art-nouveau facades are heavily decorated with stucco elements and are therefor imposing difficulties upon any model based feature detection approach because their appearance is relatively unpredictable.

Also, facades of this category contain many fine grained details and are thus very difficult to model or reconstruct automatically. There are some properties of all typical facades which are also suitable for segmenting art-nouveau buildings. Symmetries, for instance, are reliable indicators of the existence, size and location of window tiles in a very large class of building facades. In this thesis translational symmetries are used to identify repetitive features and segment the facades into tiles accordingly.

2.1.2. Overview

The algorithm takes as input a single orthogonalized view of a facade. The output is a 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 the high dimensional space of similar image regions. This is done by comparing small image regions on multiple resolutions

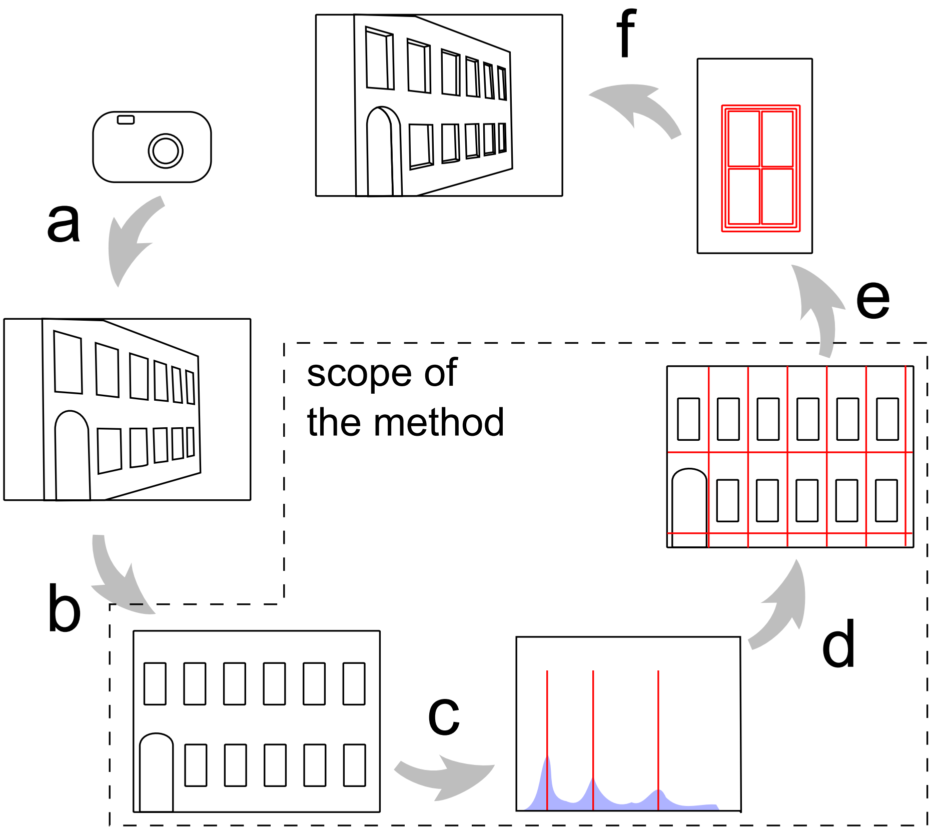


Figure 2.3.: Overview over the steps of the method and scope (represented by the dashed line) with respect to the whole urban reconstruction process. The main steps are **a)** image acquisition, undistortion, storage and efficient retrieval, **b)** triangulation of 3D points from multiple views, identification of facade planes, extraction of orthogonalized facade images, **c)** identification of dominant repetitive patterns and their offsets, **d)** segmentation of the image into repetitive tiles, **e)** exact identification and reconstruction of 3D facade details, **f)** efficient rendering of large reconstructed urban 3D data for visualization applications.

of the image for similarity. Because it is not feasibly to compare all potential corresponding regions in the image for efficiency reasons a Monte Carlo importance sampling strategy is applied to collect statistical evidence about any translational similarities. The developed multi-resolution similarity measure based on the normalized cross-correlation coefficient is shown to greatly improve the quality of the results. Further improvement is achieved by focusing on patterns that actually cover a larger image area. 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 [CM02]. The computational result of this stage are offsets in pixels which relate directly to the prevailing repetitive patterns in the image.

Localization 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. 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.

Dividing the image into facade tiles. Eventually the image is partitioned into regions with and without repetitive patterns. For the regions which exhibit repetitive patterns the most dominant pattern is selected and the pattern's 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 continuative reconstruction algorithms.

2.2. Search for dominant repetitive patterns

The goal of this stage is to search for repetitive patterns and measure their representative spatial offset. From all the detected patterns eventually the dominant ones are extracted and supplied as input for the next stage. Before going into detail about the search algorithm the next section elaborates on the understanding of repetitive patterns in the context of this thesis.

A closer look at the typical structure of facades helps to understand which image patterns are relevant for window detection. In most facades there are 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 due to their different use as shops or restaurants. Other than that, most facades exhibit strictly regular structure. If we consider a sequence of axis-aligned pixels as a function of the intensities we notice certain regular repetitions in the signal (figure 2.4). These repetitions are coherent over multiple adjacent pixel lines of the image. In conclusion, coherent axis-aligned translational similarities that are recognized over larger connected areas are the relevant patterns to be identified for detection of repetitive arrangements of similar windows or other facade elements in a typical facade image.

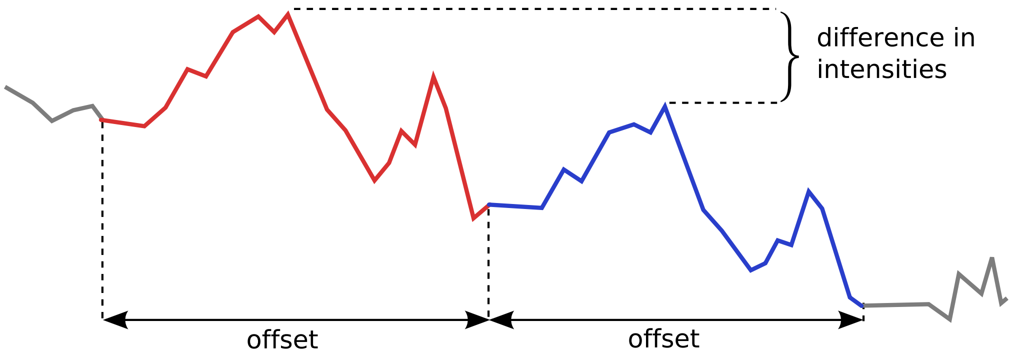


Figure 2.4.: Example of a repetitive pattern in 1D with a highly similar but not exact instance. Relative differences in signal intensities between instances of the pattern should not influence the detection algorithm. An appropriate similarity measure must be applied which is insensitive to the overall intensity level of the region.

2.2.1. Definition of a Repetitive Pattern

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 by its *offset*, the smallest distance to the next most similar recurrence of certain distinguishable features in the original sequence of the pattern which is called a repetitive instance (see figure 2.4). A signal without distinguishable features does not exhibit repetitive patterns. In the case of patterns in images the distinguishable features are relatively fast changes of the intensities. 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. Since we are interested in repetitive image features which represent 3D-details on flat facades, the characterizing features are corners and edges.

In the context of this work, a repetitive instance of a pattern is never expected to be perfectly the same but should be highly similar. An appropriate measure for the similarity of two image regions which is insensitive to relative differences in intensities and not easily disturbed by noise and other common image artifacts is presented later. The insight, that the similarity of patterns depends mostly on the discontinuous image features is later used to implement an optimized search strategy.

There may be many patterns with different similarity on different image resolutions which

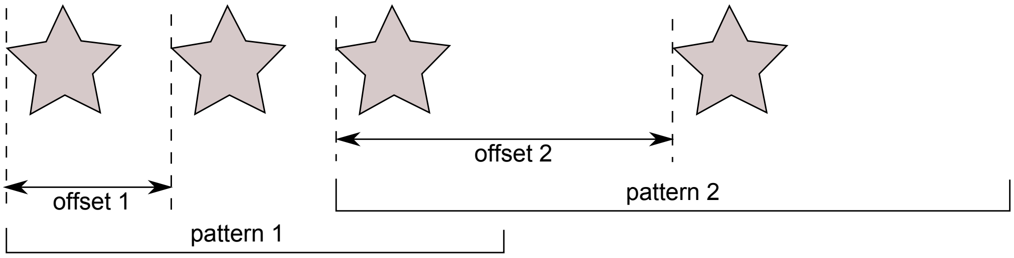


Figure 2.5.: For simplicity patterns are defined by their offset only. Accordingly in this image there are two different (overlapping) patterns. This does not hinder correct tile segmentation.

overlap each other. This makes it hard to distinguish the relevant patterns which are generated by similar architectural detail from the other patterns which are mostly random. Experiments have shown, though, that the relevant patterns for our purposes are those with the highest multi-resolution similarity.

Note that for this thesis the actual information carried by the image is not of interest and is not subject to the analysis. Just the similar re-occurances of significant instances of visual patterns is detected under the simplifying assumption that these patterns are strictly orthogonal and grid-aligned.

2.2.2. Segmentation of Repetitive Patterns

We can distinguish between repetitive and non-repetitive regions in a signal or image[1]. When traversing an image in a specific direction we can also distinguish between the first occurrence of a specific pattern and its repetitive instances. This thesis proposes an image segmentation technique which divides an image in repetitive and non-repetitive regions and splits each pattern into its repetitive instances.

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 at the offset starts to differ too much from the original instance. The bounds of a repetitive pattern are not sharp and have to be defined by a similarity threshold.

[1] Note that we only consider the image intensities of grey scale images in this work. We could consider colors with some performance hit but grey values proved to be good enough for our purpose.

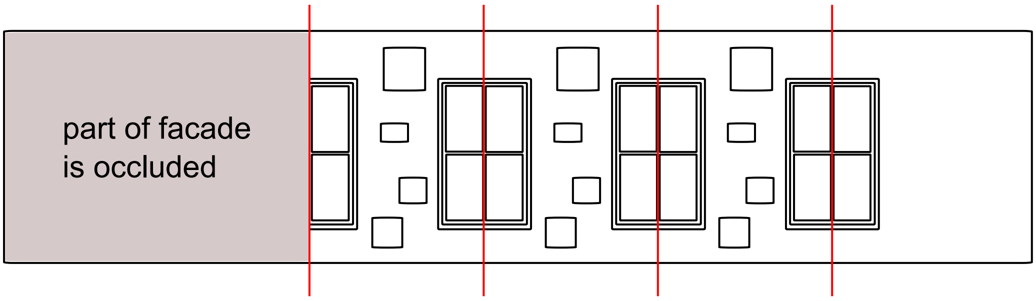


Figure 2.6.: Without a priori knowledge about the signal content it is not possible to evaluate the correctness of a split. In this case half of the first window has been occluded so the start of the repetitive pattern is shifted.

With such a threshold non-repetitive regions can be distinguished from pattern regions and a segmentation is possible.

We also want to split the original region from the repetitive instances in order to split a pattern of many repeating windows into tiles with only one window (or other facade element) each. This problem can not be solved in general without knowledge about the appearance of the pattern (see figure 2.6). From a signal processing point of view, there is no definite start or end in a repetitive pattern such as an infinite sine wave for instance. After defining a start point we can split a sine wave into periods using the offset 2π. For building facades, we can constrain our input images to complete pictures of a facade, such that it is impossible (except in case of occlusions like in figure 2.6) for a pattern to start in the middle of a window. Assuming this, a series of repetitive instances of a pattern is easily segmented by placing a splitting line at every offset pixels.

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 2.7. There are two possible concurring segmentations based on either the one pattern's offset or the other's. A solution to this problem which is applied in this thesis is to exclude some of the detected patterns according to a priori knowledge or image area constraints.

Another problem is the non-uniqueness of patterns. The same signal can be interpreted as many different patterns with different offsets. The offsets of these patterns are, of course, related arithmetically. A pattern with offset δ can be interpreted as a larger pattern with offset 2δ. This can be detected easily and eliminated, as described later. Multiple interleaved patterns with different offsets which are interpreted as the sum of different non-multiple patterns

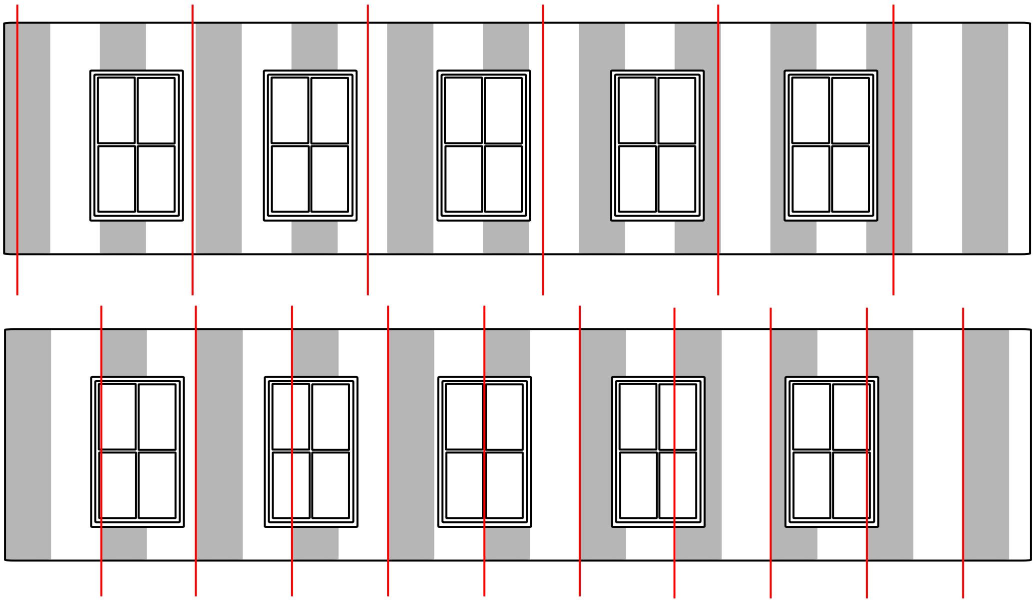


Figure 2.7.: Two overlapping repetitive patterns and their respective splitting lines. There are often highly frequent overlapping patterns, especially in art nouveau facades which exhibit much decor. The problem can be solved by constraining the offset to minimum and maximum values.

are a much more challenging problem which is not yet solved because it rarely happens.

2.2.3. Similarity measure

Repetitive patterns are defined by local self-similarities in an image. To measure the similarity of image regions we need a robust operator that is suitable for images of repeated real world objects which can exhibit a large range of defects. In literature, a common comparison operator is *cross correlation* of image patches which is simple and relatively cheap to compute making it the ideal basis for the implementation of an efficient similarity measure. To remain insensitive to relative intensity variations in the compared locations the *normalized cross correlation coefficient (NCC)* is computed. Due to its simplicity and efficiency the NCC was decided to be employed to measure similarity between distinct image locations[2].

NCC eliminates the differences in intensities caused by lighting conditions by substracting the mean of the subimages and normalizing the resulting dot product with the standard deviation.

[2] In theory other operators that are useful for template matching could be applied. In our case, NCC seems to be a good trade-off between performance and robustness.

That is, the normalized cross-correlation of a template $t(x,y)$ with a subimage $f(x,y)$ can be written as

$$NCC(t,f) = \frac{1}{n-1}\sum_{x,y}\frac{(f(x,y)-\overline{f})(t(x,y)-\overline{t})}{\sigma_f \sigma_t} \tag{2.1}$$

where n is the number of pixels in $t(x,y)$ and $f(x,y)$.

In functional analysis terms, this can be thought of as the dot product of two normalized vectors. That is, if

$$F(x,y) = f(x,y) - \overline{f} \tag{2.2}$$

and

$$T(x,y) = t(x,y) - \overline{t} \tag{2.3}$$

then we can write the operator as

$$NCC(T,F) = \left\langle \frac{F}{\|F\|}, \frac{T}{\|T\|} \right\rangle \tag{2.4}$$

where $\langle \cdot, \cdot \rangle$ is the inner product and $\| \cdot \|$ is the Euclidian Norm.

The size of the template $t(x,y)$ is called *window size* further on. The impact of window size when measureing similarity is discussed in more detail in the next section.

2.2.4. Influence of the window size

When measuring local similarities the size of the sub-image regions to compare, in short the *window size*, is an important parameter to consider with respect to performance and robustness. The Cross Correlation of small windows like 3x3 or 5x5 pixels can be computed very fast as compared to larger window sizes like 63x63 or 127x127 are very expensive to compute. This is due to the computational complexity of Cross Correlation which is quadratic with respect to 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. Small window sizes are sensitive to damages of repetitive *real world patterns* in images such as dirt, scratches, camera chip noise and other micro differences which influence the resulting values and increase the variance of the measured data. In order to gain sufficient robustness for the similarity operator it is necessary to compare sub-images of at least 15x15 pixels. This window size is, as determined empirically with facade images, the best trade-off between speed and operator robustness.

Also, when seen through a small window certain image regions may look very similar to others even if they are not when seen through a larger window (see Fig. 2.8). As a result of very small window sizes such as 3x3 pixels, small patterns with high frequencies are favored over larger patterns with lower frequencies. This is an example of overlapping patterns. As a conclusion, using a constant window size increases the potential to favor the *wrong* patterns over others or, in other words, the similarity measure is biased by the window size of its operator.

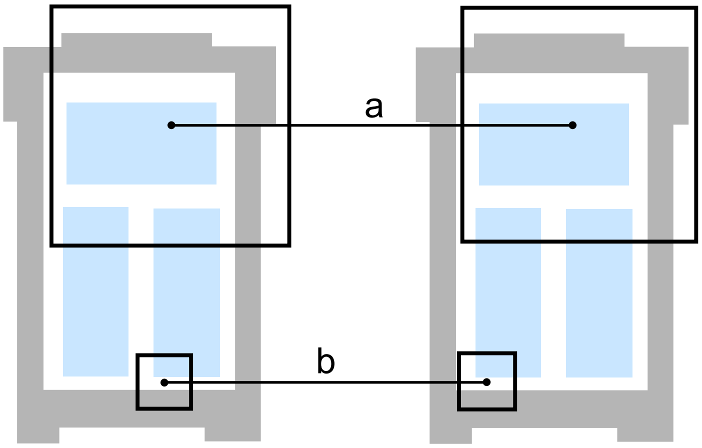


Figure 2.8.: Two differently sized similarity windows with highly similar matches. a) A correct match with a window size of similar dimension with respect to the size of the sampled features. b) An example of a wrong match with a high similarity value caused by a too small sampling window.

This imposes that an approach with a constant window size is not feasible because the size of the patterns is not known a priori. To solve this problem we need to change the definition of our similarity measure to automatically adapt to the pattern's size, in order to obtain unbiased results.

A possible approach to this problem is to choose from a set of different operator sizes at random e.g. 15, 31, 63, 127 pixels. However, due to the quickly increasing computational cost of larger window sizes a different strategy has been chosen.

2.2.5. Multi-Resolution Similarity

Rather than relying on larger windows to improve robustness and reduce bias the multi-resolution similarity operator measures similarity with a constant window size on different resolutions of the image at the corresponding location of the unscaled image. The similarity values are computed for every different resolution and then averaged to form the final multi-resolution similarity value.

This can be implemented quite straight forward with a classical image pyramid. Each layer of the image pyramid is computed by subsequently scaling down the image with the factor $s = \frac{1}{2}$, where we use cubic down-sampling to preserve smoothness. On the down-scaled image the same absolute window size covers $\frac{1}{s^2}$ -times the area as in the next lower image pyramid layer, which means, that although we have the same operator size in pixels we are comparing much larger areas. The similarity ς results from similarity in all pyramid levels which have been taken at the closest position to the original position in the unscaled picture and are then combined into the final result by taking the mean.

If f_k and t_k are the template and subimage of constant size accross all pyramid layers centered at the positions $\frac{p_i}{s^k}$ and $\frac{p_j}{s^k}$ in the k-th pyramid layer, our multiscale similarity operator between the two image positions p_i and p_j in the original image is given by

$$\varsigma(p_i, p_j) = \frac{1}{S}\sum_{k}^{S} NCC\,(t_k, f_k)\ , \tag{2.5}$$

where S is the number of layers in the image pyramid, s the scale factor for down sampling from one layer to the next higher one and NCC is the similarity operator which operates on the k-th layer of the pyramid defined over the input image I. As shown in figure 2.9 the window size is kept constant on all layers.

A very big advantage of this multi-resolution similarity measure is its robustness against any sort of highly frequent noise or variations in the image. On the other side, architectural facade features are typically evident in both high, medium and low image frequencies. This is demonstrated by figure 2.9 where the windows are still well recognizable even on highly down scaled[3] pyramid layers.

Since we are searching exactly for such repetitive features that are evident accross all pyramid layers and the noise or small variations in the patterns are only evident in high-resolution

[3] Cubic down sampling has been applied which is equivalent to a low-pass filter in the frequency domain.

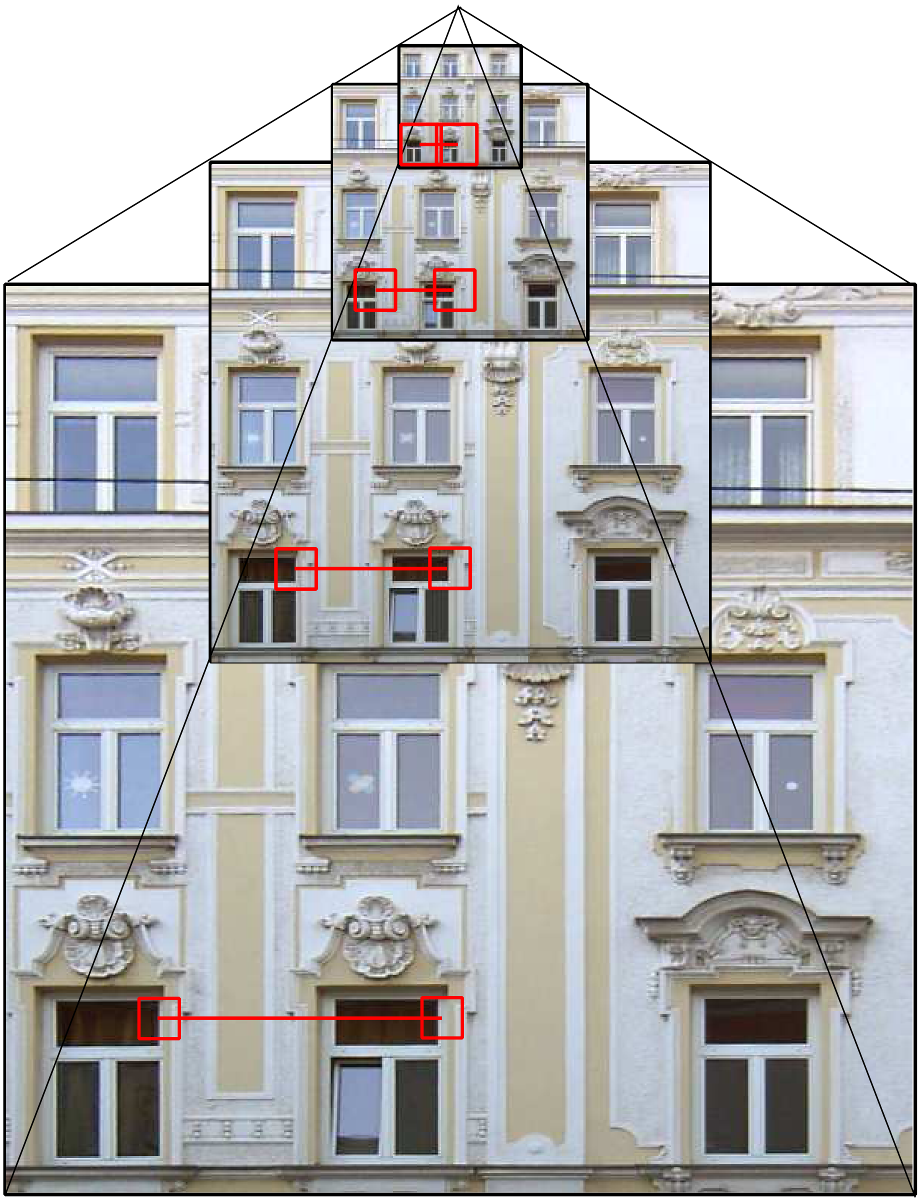


Figure 2.9.: Multi scale similarity measure.

pyramid layers multi-resolution similarity matching is conveniently robust for this kind of application compared to the simple similarity measure on the original image only. On the other hand, while being very robust, multi-resolution similarity is also more accurate than measuring on down scaled or low-pass filtered images only because it measures also the high resolution layers where small image features such as edges and corners allow exact matching of repetitive image regions. Hence, the multi-resolution similarity operator produces high quality results[4].

By using a constant window size the multi-resolution similarity operator on the image pyramid is highly efficient compared to using large similarity windows on the original image. Also, the multi-resolution similarity operator on the pyramid is not totally equivalent to the multi-sized similarity operator on the original image because the latter lacks the implicite low-pass filtering of the former approach.

Also, by virtually comparing relatively large areas by using small ares of low-resolution pyramid layers of the image the quality and the performance of the multi-resolution similarity measure is completely independent from the size of the input image and the sizes of the repetitive patterns we are interested in.

The image pyramid is computed only once so it does not add to the complexity of the method. Given that the number of pyramid layers is bound and will not be higher than five to ten layers depending on the size of the original image (possible image sizes are bound too) this also does not add to the algorithmic complexity of the multi-resolution similarity operator. Of course, the higher the pyramid the more computations have to be made for each similarity value. This added computational cost of evaluating the similarity operator on multiple pyramid levels can be greatly reduced by a so called *early break strategy*. Early break stops the evaluation of all pyramid layers if the similarity value on the highest pyramid level (lowest resolution) is under a certain threshold. In practice a very high number of compared regions are not similar at all. With early break the full cost of evaluating the similarity operator on multiple resolutions can be reduced to a single measurement in all the cases where the compared windows do not match and thus a high quality multi-resolution evaluation is not necessary anyway.

[4] Multi-resolution image analysis is a common approach to robust matching, e.g. [ZG02].

2.2.6. Monte-Carlo self-similarity sampling

This section discusses, how the self-similarities of an image can be efficiently and sufficiently determined by a stochastic sampling process. The basic idea is, that if enough randomly chosen image locations are compared for similarity locally the offsets of the relevant repetitive patterns in the image can be determined through a statistical analysis of the sampled global distribution of offsets.

A randomly chosen pair of image locations are considered similar or repetitive if the similarity is beyond a certain threshold or if the similarity value is higher than for any other pair on the same line when one of the pair's location is fixed. The distance of the positions of such a pair of corresponding image locations with a high similarity value is called the *offset* of the local repetitive image pattern. From all the randomly sampled offsets with high similarity we can select the most prevailing offsets. These represent the dominant repetitive patterns in the image. The following paragraphs go further into detail.

Exhaustive search. Before we look at the efficient way to solve it, let's first take a naive approach to better understand the problem. It is quite a computational challenge to search for all the repetitive patterns originating from arrangements of similar image regions because we are searching a very large domain of many possible combinations of offsets and positions. A naive deterministic approach to the problem of finding horizontal translational similarities would be the **brute force** approach also known as **exhaustive search**:

1. computing and storing similarity values ς for all possible horizontal offsets Δ in a range D as subset of the image width W:

   $$H(i) = \varsigma\left(p_i, p_{\Delta_j}\right) \ ,$$

   where $\forall\, i \in I$ and $\forall\, j \in D \subseteq \{W\}$.

2. statistical analysis of the computed data H.

This is an approximation of the well known approach of extracting maxima in the auto-correlation surface of a repetitive image which is used in many publications to repetitive pattern recognition. The standard procedure is to compute the auto-correlation of the complete image and to employ different techniques to analyze it [HLL01], [LCT04], [HLEL06]. Another exhaustive search approach which is specialized on facades compares large image slices using Mutual Information [MZWVG07]. All these methods are very costly operations especially for high-resolution images supplied by common digital cameras. Also, for the objective of this thesis calculating an auto-correlation surface of the whole image is not necessary. Since we are interested in horizontal and vertical similarities only, a majority of the computations of the auto-correlation surface would be wasted for nothing. The auto-correlation surface is actually an implementation of the classical brute force search strategy when it is used to search for repetitive patterns. Since exhaustive search is too costly especially for large high-resolution images a more sophisticated search strategy is presented in the next paragraph.

Monte Carlo sampling. A common approach to deal with complex or high-dimensional search spaces are Monte-Carlo (MC) solutions. Using MC sampling to obtain samples of the same data as computed above 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 statistic probe of the similarity at a constant number of random positions:

1. computing and storing N similarity values ς for all possible offsets Δ in the range D as subset of the image width W, where i is determined randomly from an appropriate distribution and $\forall j \in D \subseteq \{W\}$:

$$H(i) = \varsigma\left(p_i, p_{\Delta_j}\right) \; .$$

2. statistical analysis of the computed data H

The quality of the results of Monte Carlo sampling is of course depending on the probe size (the number of random samples taken) so eventually with a very large probe size the Monte Carlo result converges against the true solution. If some information about the underlying distribution that is sampled is known the convergence speed of the Monte Carlo process can be sped up dramatically. With clever sampling strategies that make use of these informations the results of Monte Carlo sampling quickly converges against the true deterministic solution with relatively few samples.

The big advantage of Monte Carlo solutions is that they are much more efficient to compute, especially for high dimensional problems. Indeed, in our case the computation of the most important offsets is significantly faster than the equivalent exhaustive search solution.

2.2.7. Importance sampling

Importance sampling is a technique to improve the convergence speed of Monte Carlo sampling by incorporating *a priori* knowledge about the sampled distribution into the sampling process. In contrast, simple sampling just casts uniformly distributed random samples. The sampling error with simple sampling is much higher than with importance sampling for certain functions.

We want to sample the distribution of repetitive image features which comes down to the distribution of pairs of image locations that are highly similar. In this case, importance sampling would be to draw more samples from some image regions than others based on information where such repetitive regions might be. As a result, the regions of interest are sampled much better than others and thus the number of necessary samples is kept as low as possible.

Facade elements such as windows, balconies etc. are characterized by sharp orthogonal edges and corners. Based on this information we can implement an importance sampling strategy. It is not so important to sample image regions without any edges or corners because they might not contain any facade elements. Instead we focus on edges and corners which are better indicators for facade elements. The implementation of such an edge-based importance sampling strategy is quite simple: in a pre-processing step an edge image is computed using Sobel-filtering and Canny edge detection [Can86]. Then the positions of all edge pixels are collected into an array. During the stochastic sampling process the Monte Carlo algorithm takes positions from this array at random. Using this sampling strategy, the accuracy of the result is much higher than for simple uniformly distributed random position sampling of the image while requiring significantly less samples.

Our sampling strategy can be further improved. When looking at typical Art Nouveau facades in Vienna one might notice that decorative elements which produce dominant horizontal image features are very common (see figure 2.2.7). Searching for repetitive horizontal patterns on such features would introduce a lot of errors because any offset would yield high similarity values. By adjusting the importance sampling strategy we can further improve the convergence speed and reduce the variance of the results. One way to adjust the sampling strategy would be to avoid sampling on horizontal edges. Since windows are still represented good enough by vertical edges we tune our importance sampling strategy to perform better on Art Nouveau facades by restricting random samples to positions from vertical edges only.

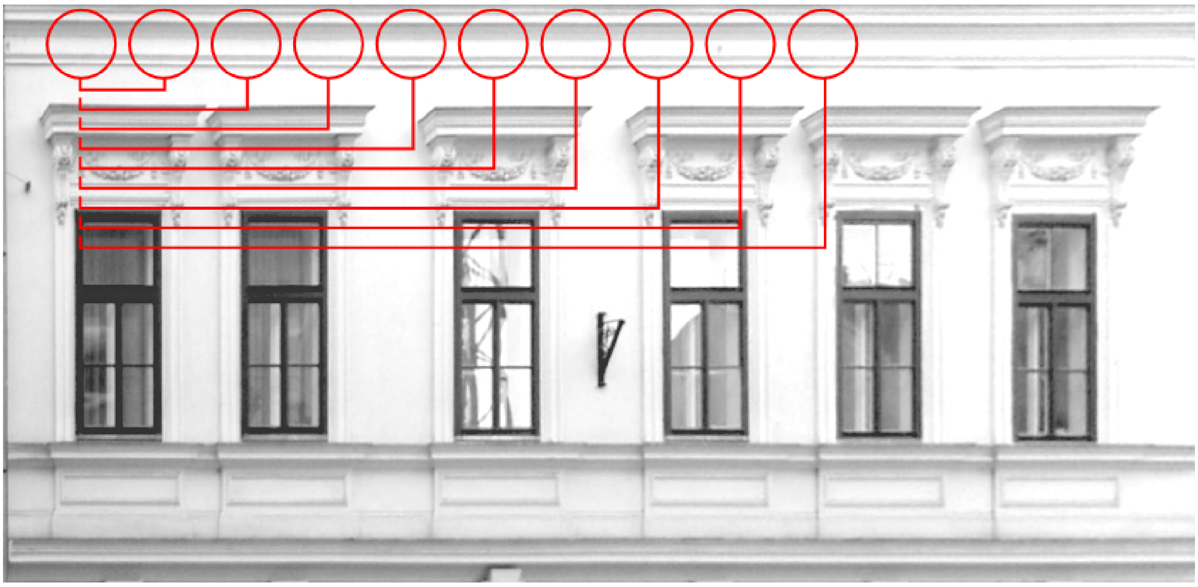


Figure 2.10.: A typical Art Nouveau facade in Vienna featuring strong horizontal edges. Such image features introduce errors when searching for horizontal repetitive patterns which results in a higher number of samples to be cast to get reliably results.

Distinguishing important patterns via estimated image size. The offsets found through stochastic sampling need to be qualified for their relevance. The most important criterion is of course the similarity. A pattern that is based on sampled corresponding image locations with a specific offset with lower similarity values is less relevant than a pattern with a different offset and higher similarity values. However, high similarity values do not necessarily make an offset representative for a dominant image pattern. It could be just coincidence that a single pair of image regions are highly similar but do not belong to the prevailing repetitive pattern in that image.

The area covered by windows and their surrounding decoration on typical apartment building facades in Vienna ranges from 40% to 90%. This has been determined empirically from hundreds of art noveau facade photos taken on different locations in vienna. Because we are especially interested in facade features like windows, we are searching for patterns that cover a quite large area of the image. This way we are able to distinguish between relevant and irrelevant regular patterns which otherwise can not be distinguished by their similarity values.

In order to be able to estimate the image area covered by a specific pattern the random samples which are not rejected are classified and sorted into a histogram by their offset. The more samples with the same offset are counted the more area is covered by the pattern and thus the higher is its relevance. This technique is inspired by Monte Carlo integration which counts random samples to estimate the area defined by a mathematical function to compute its integral. Monte Carlo integration is a very efficient way to numerically solve complex integrals especially for high-dimensional functions. It is also known to outperform standard numerical quadrature methods with respect to accuracy if a sophisticated sampling strategy is applied, such as importance sampling.

The proposed sampling process to identify large image patterns casts a number of random samples and sorts the resulting offset into histogram bins if they meet certain criteria. As said before, patterns that cover more image area are more relevant than others so the quantity of occurrence of a specific offset is the decision criterion. Larger patterns will cause significant peaks in the offset histogram, because according to Monte Carlo integration, the larger an area the more random samples will hit it. The resulting histogram represents the distribution of similar offsets in the image.

2.2.8. Sample quality criteria

In order to identify patterns and measure their offset, we need some criterion to judge what is the best matching corresponding region for a given location. A trivial criterion is the similarity value itself. This criterion is called the **threshold criterion**.

The *threshold criterion* simply defines a global threshold for the accepted similarity values. The similarity values range of our operator is between 1 and -1, where 1 means high translational similarity and ≤ 0 means no translational similarity at all. For identification of translational repetitions we could define a threshold criterion with a similarity threshold of 0.8. This means, that all samples with a similarity value below 0.8 are rejected and thus treated as not significantly similar.

The histogram classification function $h(\Delta)$ with threshold criterion for N random samples and threshold t is given by:

$$h(\Delta) = \sum_{i}^{N} \begin{cases} 1 & \text{if } \varsigma(p_i, p_\Delta) > t \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

The 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. When measuring similarity with the normalized cross-correlation operator, a threshold of 0.8 or higher 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 a lot of 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. ´

A much more accurate criterion for finding the best recurrence of a spot in the image is to compare the similarity values of multiple possible candidate offsets and choose the best match. This **best match criterion** is shown to significantly boosting the signal-to-noise-ratio of the estimated offset distribution histogram. Basically, 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:

$$h(\Delta) = \sum_{i}^{N} \begin{cases} 1 & \text{if } \Delta = \Delta_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

where $\Delta_j = \arg\max \varsigma\left(p_i, p_{\Delta_j}\right), \quad \forall \Delta_j \in D$. The range D is defined as a subset of all possible offsets W: $D \subseteq \{W\}$ 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.

2.2.9. Extraction of the relevant patterns

Due to real imprecisions of the facade or due to the unavoidable perspective distortions on photographs the best match of different samples of the same pattern might show some small

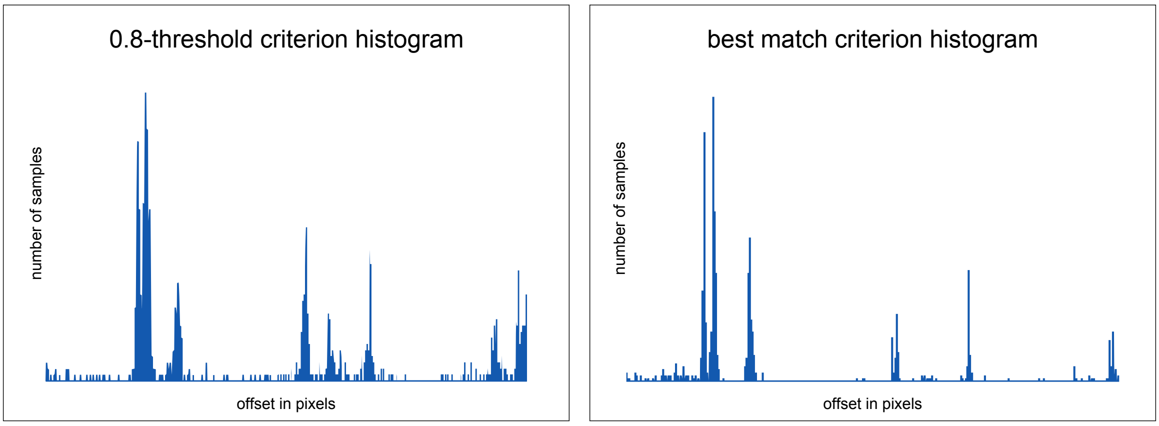


Figure 2.11.: Comparison of histograms resulting from 100k samples with *threshold criterion* selection (left) and 1k samples with *best match criterion* selection (right). The broad peaks in the left hand histogram and high peaks of irrelevant offset combinations are signs of the much higher overall error of the simple *threshold criterion*.

differences. Even after rectification certain artifacts resulting from perspective distortions remain in the image which account for variations of the measured offsets of the same pattern in different image regions. 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 which might corrupt the results unless a good evaluation method is used. As a result, a technique is required to classify similar significant peaks of offsets into a cluster representing the same repetitive pattern. In other words, the significant peaks need to be extracted from the histogram using a method that is not disturbed by the unavoidable noise.

To reduce the impact of noise it is advantageous to smooth the histogram function before peak extraction. The histogram curve can be smoothed with a box filter or a Gaussian kernel or any other smoothing operator. In the reference implementation a simple and efficient smoothing term has been chosen. Each value $f(x)$ of the sequence f is normalized such that

$$f(x) = \frac{1}{2n+1} \sum_{k=-n}^{n} f(x+k) \tag{2.8}$$

where n is the size of the smoothing kernel. Note that this smoothing kernel directly alters the sequence it operates on instead of creating a filtered copy of the sequence. This means that previously filtered values are re-used which results in a error-diffusion of the noise. The major peaks which mark the relevant offsets remain while the noise and the outliers are removed.

The results of this smoothing operator are quite similar to a Gaussian blur but come at lower computational cost.

Smoothing of the histogram curve is essential for the peak extraction. If the ambient noise generated by the Monte Carlo random sampling algorithm is not smoothed out the peak extraction would find too many peaks. In this context it is also important to discuss 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 therefor 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 size of $n = \frac{d}{50}$ proofed to be useful for most images, where d is the currently relevant image dimension (with or height), dependent of the processing direction.

In order to preserve precision of procedure while smoothing the histogram caution is advisable not to smooth to much. In an extensively smoothed histogram near standing peaks may merge into a single peak that is located in the middle. The reference implementation smoothes only once with the above given smoothing window size.

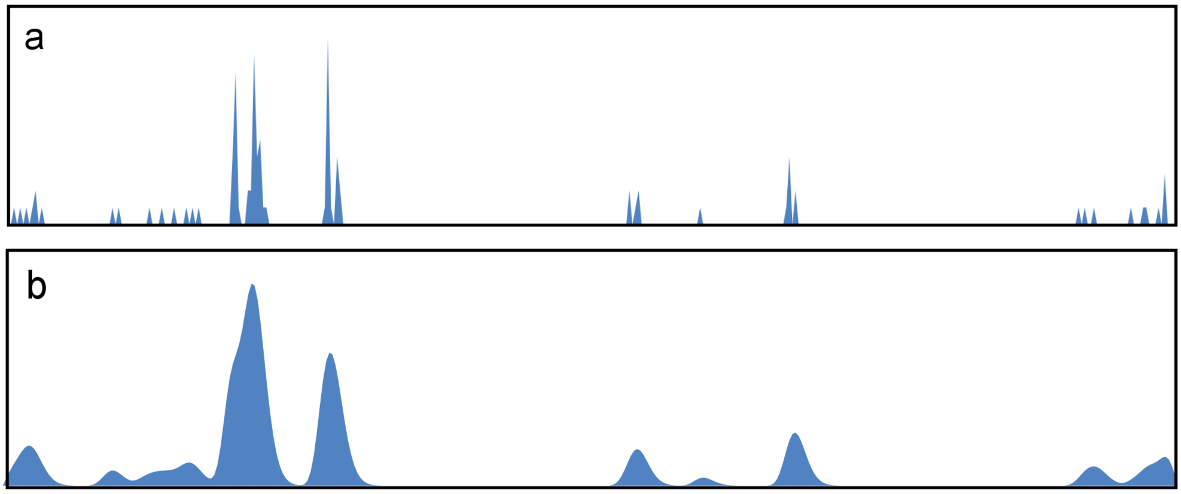


Figure 2.12.: Original histogram (a) and a smoothed and normalized histogram (b). In the smoothed histogram some close peaks are merged together because of oversmoothing. This reduces the number of concurring extracted peak locations on the one hand but also degrades precision of the segmentation on the other hand.

The peaks can either be obtained from the smoothed histogram function by extracting the maxima by numerical differentiation or better by clustering. The problem with maximum extraction is, that even very small variations may result in local maxima which are not relevant peaks. One could try to define a threshold for relevant maxima by means of the standard deviation but it is error prone. The best method for robust peak extraction proved to be *mean shift clustering* [CM02]. The mean shift algorithm is a nonparametric clustering technique which does not require prior knowledge about the number of classes and does not constrain the shape of the clusters.

The mean shift algorithm is an iterative procedure where a number of sampling windows (i.e. circles with a specific radius) successively move towards the greatest increase in density and finally stabilize in the center of mass of a cluster. For every iteration the center of mass for every window is computed and the window is moved to that new location. The vector from the old location to the new location (the *mean shift vector*) always points towards the direction of maximum increase in density. The iterative procedure is guaranteed to converge to a local density maximum. The number of resulting clusters depends on the window size and the position and number of starting values.

Figure 2.13.: Position of mean shift clusters (red) after a few iterations over a smoothed offset histogram.

Post processing of extracted offsets. When stochastically sampling repetitive patterns that are normally re-occurring multiple times in the image, it happens that a very good match or even the best match is not the first recurrence of the pixel configuration in the image. According to that, 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 constantly spaced through 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 of the histogram in figure 2.15 and the corresponding annotations in figure 2.14 for examples of multiples in a facade image.

A simple but very efficient partial solution to this problem is to remove all offsets which are close to integer multiples of the smallest offsets. The downside of this approach is, that in

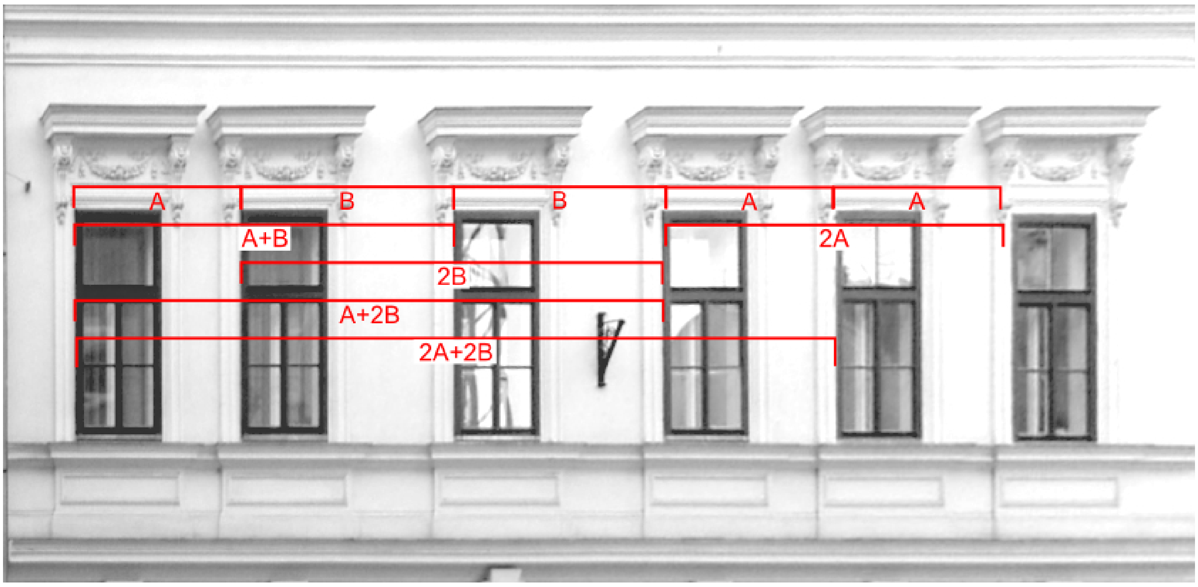


Figure 2.14.: Demonstration of a number of possible multiples of offsets A and B which might obscur the results of the histogram extraction. Two, three and four times multiples of an offset happen quite often and can be easily removed by postprocessing.

some cases relevant offsets have been removed because they were close to multiples of a smaller offset. For the goals of this thesis, though, this solution proved to be feasible enough. For future work, there might be room for potential improvements regarding this matter. For instance, the fact that a highly similar match may not be the first recurrence of a pattern could be incorporated into the sampling process by prioritizing samples which are repeated more often with a high similarity in one direction.

Another option is to dramatically reduce the number of possible multiples by selecting only images with strong perspective, as described in the next paragraph.

The effect of perspective distortion on correctness. Tests have shown that the offset extraction works better on (rectified) facade images that have been taken from pictures with stronger perspective. On images with strong perspective distortions the next adjacent repetitive facade detail appears much more similar than any repetitive instance that is farther away. Take a look at figure 2.16 for a visual explanation.

The reason is that when the perspective facade is unprojected to obtain an orthogonalized view of the facade plane any other geometry that is not on the facade plane remains more or less perspectively distorted depending on the direction and offset from the facade plane. Thus the windows *a* and *b* from 2.16 are more similar than the windows *a* and *c*. This means

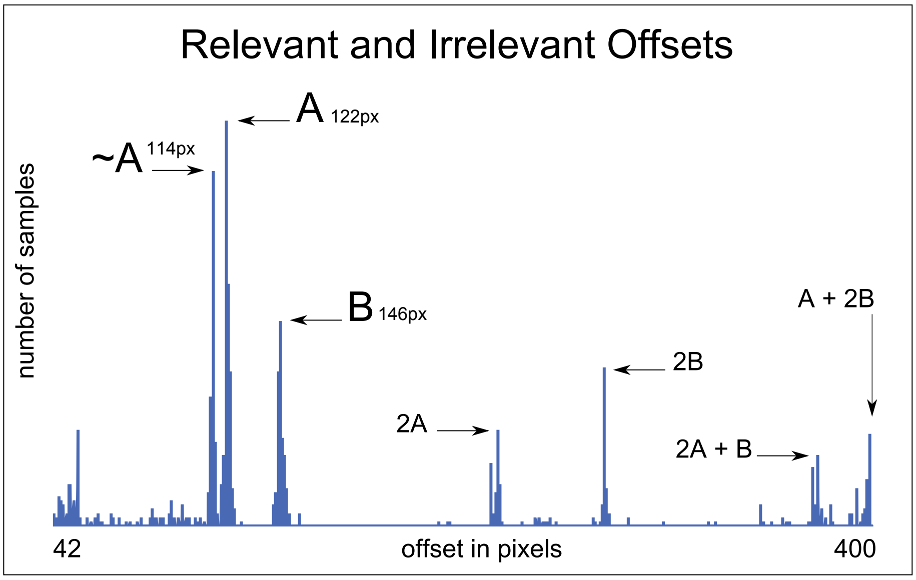


Figure 2.15.: Histogram from 1000 random samples gathered using the *best match criterion* for the facade image shown in 2.14. The multiples of the main offsets A and B can be clearly observed as secondary peaks in the histogram.

that samples of the shortest repetitive instance yield a higher similarity value than samples of any other recurrence. This means that pictures with strong perspective distortion ensure better results due to significant reduction of multiples due to the preferrance of the shortest recurrence of a repetitive pattern's instance.

2.3. Segmentation of the Repetitive Instances

We now know which patterns (given by their representative offset) are the prevailing ones in the image. We identified them by prioritizing patterns with offsets that occur significantly more often than others and exhibit reasonably high multi-resolution similarity values. Now that we have this information we would like to determine the location of each distinct repetitive pattern and its extent in the image. This allows us to reach our goal which is to create a partition of the image where every partition represents a repetitive tile of a single dominant pattern if there are any repetitions in that area of the image at all. The end result should be a partition of the facade image into floors where every floor is divided into the tiles of windows. This partition then serves as input i.e. for more sophisticated 3d-reconstruction algorithms which is beyond the scope of this thesis.

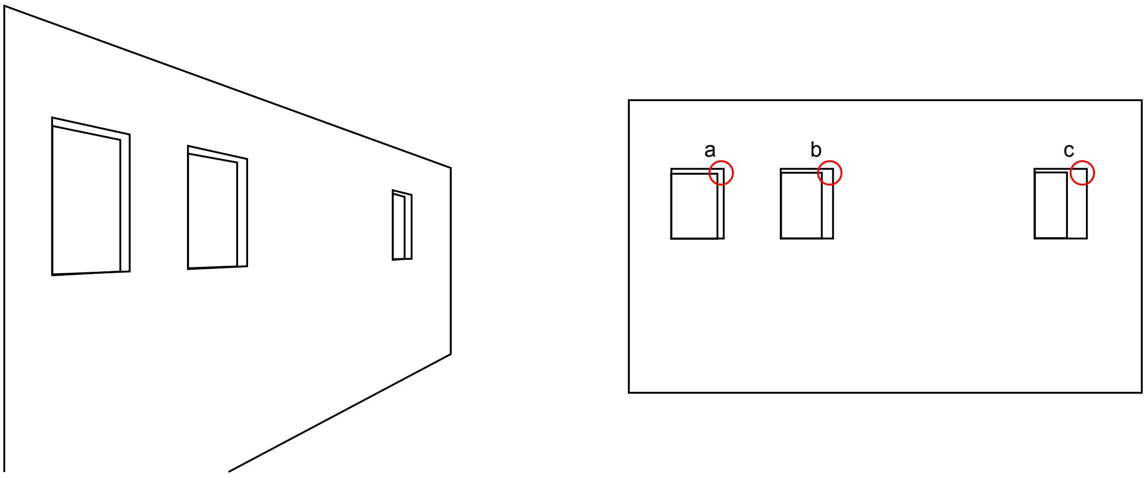


Figure 2.16.: Geometry other than the unprojected and orthogonalized facade plane remains distorted in the rectified image. This results in higher similarity values for the nearest instances of a repetitive pattern. The illustration exaggerates the effect for clarity.

2.3.1. Localization of the Patterns

The basic idea of the following pattern localization procedure is to test for each of the dominant patterns how they are distributed over the area of the image. Of course the decision whether or not a pattern prevails in a specific image region is not always a binary choice. But when compared to the other detected dominant patterns a decision can be made which of the patterns is the best fit in a specific image location.

The Similarity Curve. Of course it would be too expensive to check every single pixel in the image which of the identified patterns is most dominant so we again resort to an estimation using random sampling. The same multi-resolution similarity measure as used in the identification step serves as criterion for the relevance of a specific pattern in a specific region. For every different offset the sampled data can be seen as a curve containing the similarity values for every pixel row y or pixel column x in the image. This *similarity curve* $S(x,\Delta)$ for every dominant offset Δ based on the similarity measure ς is formally described as:

A horizontal *similarity curve* $S(x,\Delta)$ for an offset Δ is defined as follows: Using the multi-resolution similarity measure ς 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:

$$S(x,\Delta) = \frac{1}{N}\sum_{i}^{N} \varsigma\left(p(x,y_i), p_\Delta(x,y_i)\right). \tag{2.9}$$

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 reader might notice, that this sampling process is gathering positional data that actually (at least partially) could be recorded in the identification sampling process. It might even be possible to integrate the localization step with the identification step to gain even more performance. Such optimizations seem to be promising subjects for future work.

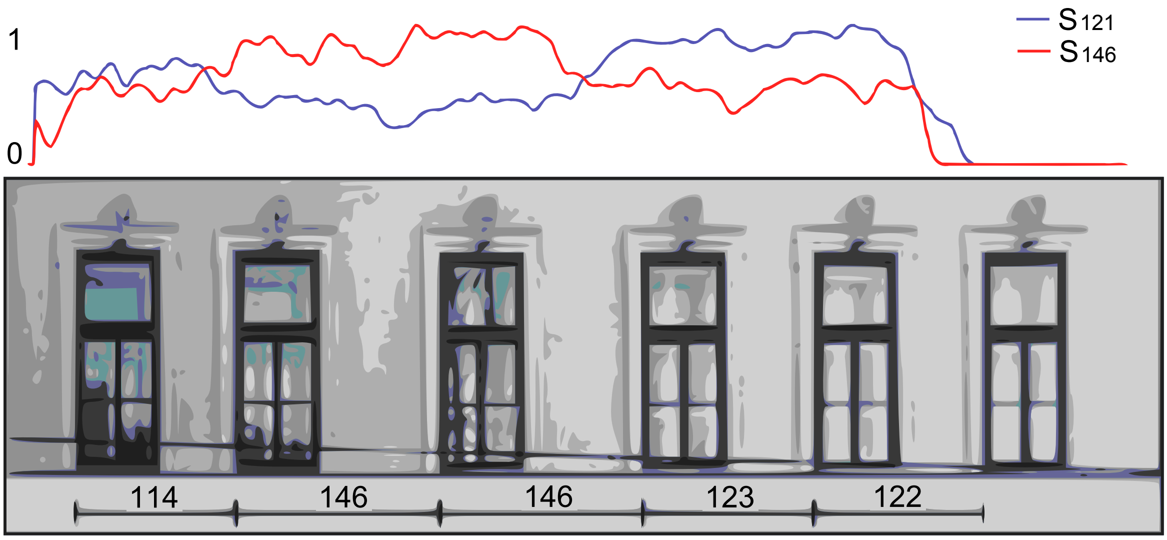


Figure 2.17.: Shows the similarity curves for the two extracted major offsets. Note that the similarity curve of the offset that is closer to the local underlying pattern is significantly higher.

The resulting similarity curves for every relevant offset are shown in figure 2.17.

With help of the similarity curve for each identified relevant offset, localization of the patterns is relatively simple. It can be done by comparing the similarity curves for each relevant offset against each other. By setting the curves in relation to each other, a decision can be made which image regions "belongs" to which pattern. Moreover, regions with very low similarity response to all major offsets are considered to be non-repetitive image regions. But first we need to divide the image into regions.

2.3.2. Segmentation

The goal of this work is to divide a rectified image of a building facade into tiles of repetitive facade elements. Using all the collected evidence, we can now carry out this segmentation task and divide the image space into tiles. As mentioned before the idea is to compare the similarity curves against each other.

Segmentation by Maximum-Projection of Similarity Curves. A simple approach would be to assign every image column the offset for which the similarity curve is highest in that specific row. Such a per-column maximum projection method has the drawback that the image might be over-segmented in areas where two very similar offsets exhibit high alternating similarity response. This method was the original segmentation approach but turned out to be not robust enough. Therefor another method has been developed as described in the next section.

Segmentation by Iterative Monte-Carlo Integration. Actually we would like to constrain the resulting repetitive regions to a plausible minimum size. The offset of the prevalent repetitive pattern is a good minimum constraint. In other words, a region that contains an instance of a repetitive image feature should be exactly the size of the offset of the feature to its next instance.

This 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: ´

$$L_{i+1} = L_i + \arg\max_{\Delta} \left(\frac{\sum_{x=L_i}^{L_i+\Delta_j} S(x,\Delta_j)}{\Delta_j} \right) \tag{2.10}$$

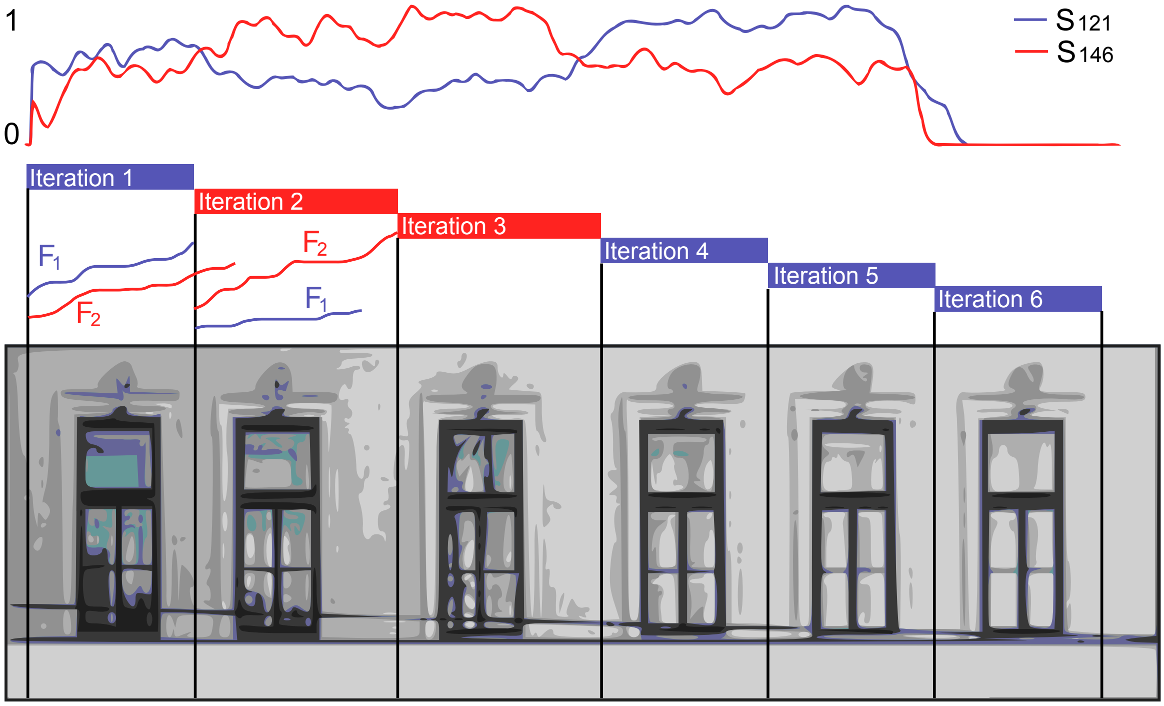


Figure 2.18.: Illustration of the iterative segmentation algorithm. For each iteration and each major offset an integral F_i of the similarity curve S_i is calculated. Since the integration is over a different range for every offset, the resulting areas are normalized to allow a comparison. The offset 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 Δ_j are the relevant offsets that have been extracted from the image. L_0 is initialized to 0 or to the first row / 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 $N = 2,3,4...$ 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 and normally their similarity is quite equal yielding unstable results depending on the random numbers used for sampling. In a previous section a possible solution to this has been discussed, namely explicit removal of "nearly non-fractional" multiples of smaller offsets. This is not the finest approach and it may lead to other errors in certain situations. A much more elegant solution is to modify the splitting function in order to slightly prioritize smaller offsets over larger ones with a weighting factor.

$$\omega(\Delta_j) = 1 - \left(\frac{\Delta_j}{\min \Delta} \varepsilon \right) \tag{2.11}$$

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

$$L_{i+1} = L_i + \arg\max_{\Delta} \left(\frac{\sum_{x=L_i}^{L_i+\Delta_j} S(x, \Delta_j)}{\Delta_j} \omega(\Delta_j) \right) \tag{2.12}$$

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.

Chapter 3.

Results

In this chapter various aspects of the proposed facade segmentation method are examined and presented. The given numbers and the discussion of the limitations of the approach should allow to conduct an objective judgement with respect to quality, correctness, robustness and performance of the method and its current reference implementation.

3.1. Performance

The performance of the method depends largely on the sampling criterion that is applied. The performance of the best-match-criterion is directly proportional to the number of pixels in the image, whereas the performance of the threshold criterion is quite constant considering that the number of samples taken (i.e. the probe size) is constant (see figure 3.1).

3.1.1. Best-match vs. Treshold criterion

The following table summarizes horizontal segmemtation 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 aequivalent to horizontal segmentation.

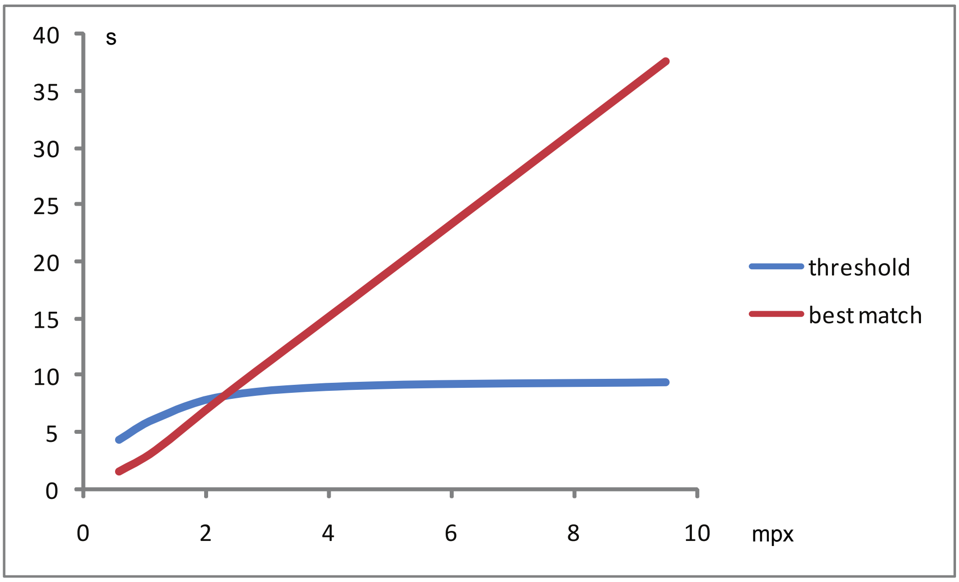
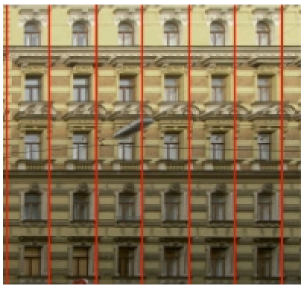
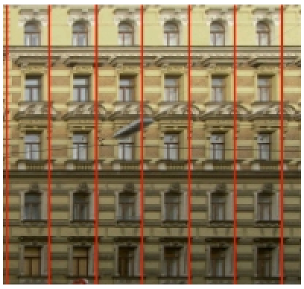


Figure 3.1.: Performance comparison of the sampling criteria "best match" versus "threshold". The graph displays the running time of each sampling strategy as a function of image size.

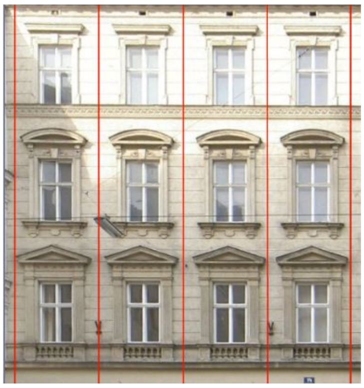
<table>
<tr><th colspan="4">Performance of best match criterion</th></tr>
<tr><td rowspan="6"></td><th>megapixel</th><th>time (s)</th><th>correct segmentation</th></tr>
<tr><td>0,59</td><td>1,53</td><td>yes</td></tr>
<tr><td>1,19</td><td>3,41</td><td>yes</td></tr>
<tr><td>2,37</td><td>8,49</td><td>yes</td></tr>
<tr><td>4,75</td><td>18,15</td><td>yes</td></tr>
<tr><td>9,50</td><td>37,39</td><td>yes</td></tr>
<tr><th colspan="4">Performance of threshold criterion</th></tr>
<tr><td rowspan="6"></td><th>megapixel</th><th>time (s)</th><th>correct segmentation</th></tr>
<tr><td>0,59</td><td>4,32</td><td>yes</td></tr>
<tr><td>1,19</td><td>6,33</td><td>yes</td></tr>
<tr><td>2,37</td><td>8,33</td><td>yes</td></tr>
<tr><td>4,75</td><td>9,23</td><td>yes</td></tr>
<tr><td>9,50</td><td>9,50</td><td>yes</td></tr>
</table>

The timings were recorded on a a Intel dual-core 2.4 GHz computer. The performance comparison shows the linear complexity of best-match sampling as opposed to 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 precision is required i.e. for images where the distance of different patterns which should be distinguished is relatively low.

3.1.2. Impact of random sampling probe-size on performance

The performance of this segmentation method is not only dependent on the image size but also depends on the number of samples taken.

The following table shows the horizontal segmemtation performance of a typical facade image with a resolution of 0.4 megapixels and different numbers of samplings. For the threshold sampling criterion a threshold of 0.8 was used.

	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			

These timings where recorded on a Intel dual core 2.4 GHz machine. The above table shows, that the execution time is directly proportional to the number of samples taken for any of the applied sampling criteria. Most relevant for the algorithms performance, however, is the sampling criterion and the image size, because the number of samples does not need to vary and hence is to be considered as a constant factor.

3.1.3. Complexity

The algorithmic complexity of this method depends on the sampling criterion. 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 from 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 therefor 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.

3.1.4. Parallelization

The algorithm is easily parallelizable in many different ways to take full leverage of the computation power of contemporary multi-core processor architectures. For instance, one could dived up 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 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.

Another possible way to parallelize the computation load on multi-core or multi-processor architectures is to distribute the multi-resolution Cross Correlation computations such that the computation on each different pyramid layer is executed by another processing unit. This, however, requires to sync the threads for each sample when the similarity values on different scales are averaged. This approach makes more sense for a parallel pipeline architecture such as a modern graphics GPU than for a multi-core CPU.

Another way is to use an image-processing library that is highly optimized for parallel execution on multi-core systems. For instance, the reference implementation relies entirely on the Intel Performance Primitives library for parallelization of the NCC operator. The following table demonstrates the effect of dual core parallelization on the execution time of the algorithm.

cores	time (s)
1	13.6
2	6.6

The results show clearly that allowing the algorithm to run on two cores yields the expected performance gain by reducing the execution time to a half of the time needed on a single core. The above times where recorded while taking 100 best-match samples from a 0.7 megapixel image on a Intel dual core 2.4 GHz machine.

3.2. Quality

3.2.1. Precision

The precision of the segmentation method presented herein 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 in percent of the exact solution.

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

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 are 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 is growing proportional to the absolute size of the patterns.

3.2.2. 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.

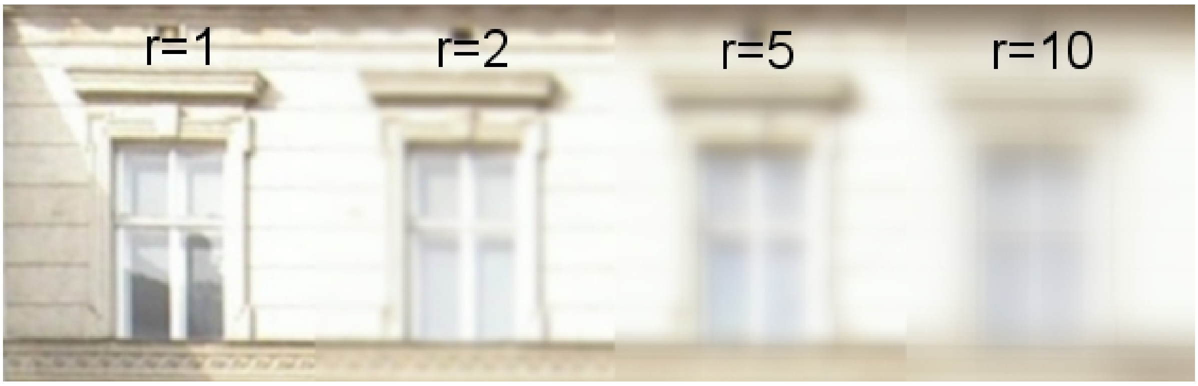
The method is parameterless with respect to image dimensions because all parameters can be defined relative to image dimensions. The advantage of such an approach is that the algorithm automatically adapts to the resolution of the input image and yields correct results without tweeking any parameters. Also, parameters can be tweaked once and will work for any input resolution.

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. Anyway, 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.

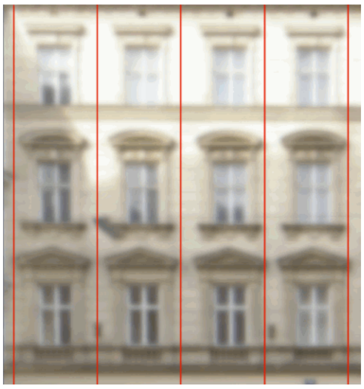
3.2.3. Robustness

The presented facade segmentation algorithm is especially robust. 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.

Robustnes against Gaussian blur. The following table compares the robustness of the best-match sampling method against the threshold method with respect to blurriness.



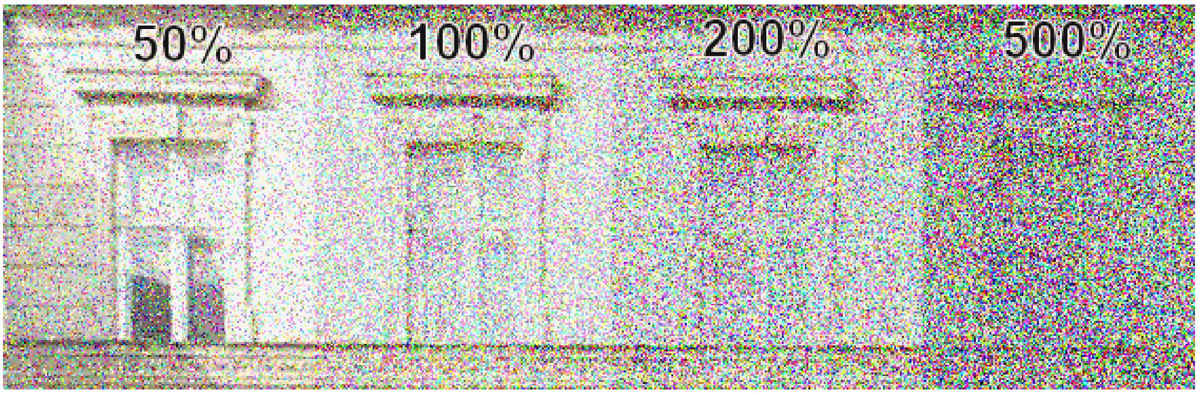
This image shows a detail of the test image under Gaussian blur with different radius.



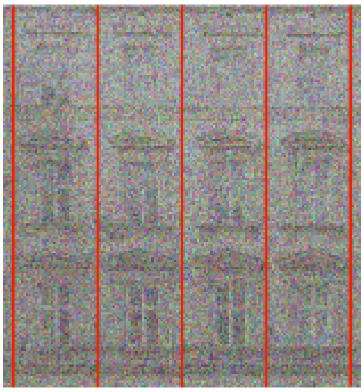
radius	best-match correct	threshold correct
0	yes	yes
1	yes	yes
2	yes	yes
5	yes	no
10	no	no

Under extreme blurring the importance sampling strategy fails and eventually 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 to be quite robust against blurriness.

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



This image shows a detail of the test image under increasing levels of random noise.

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

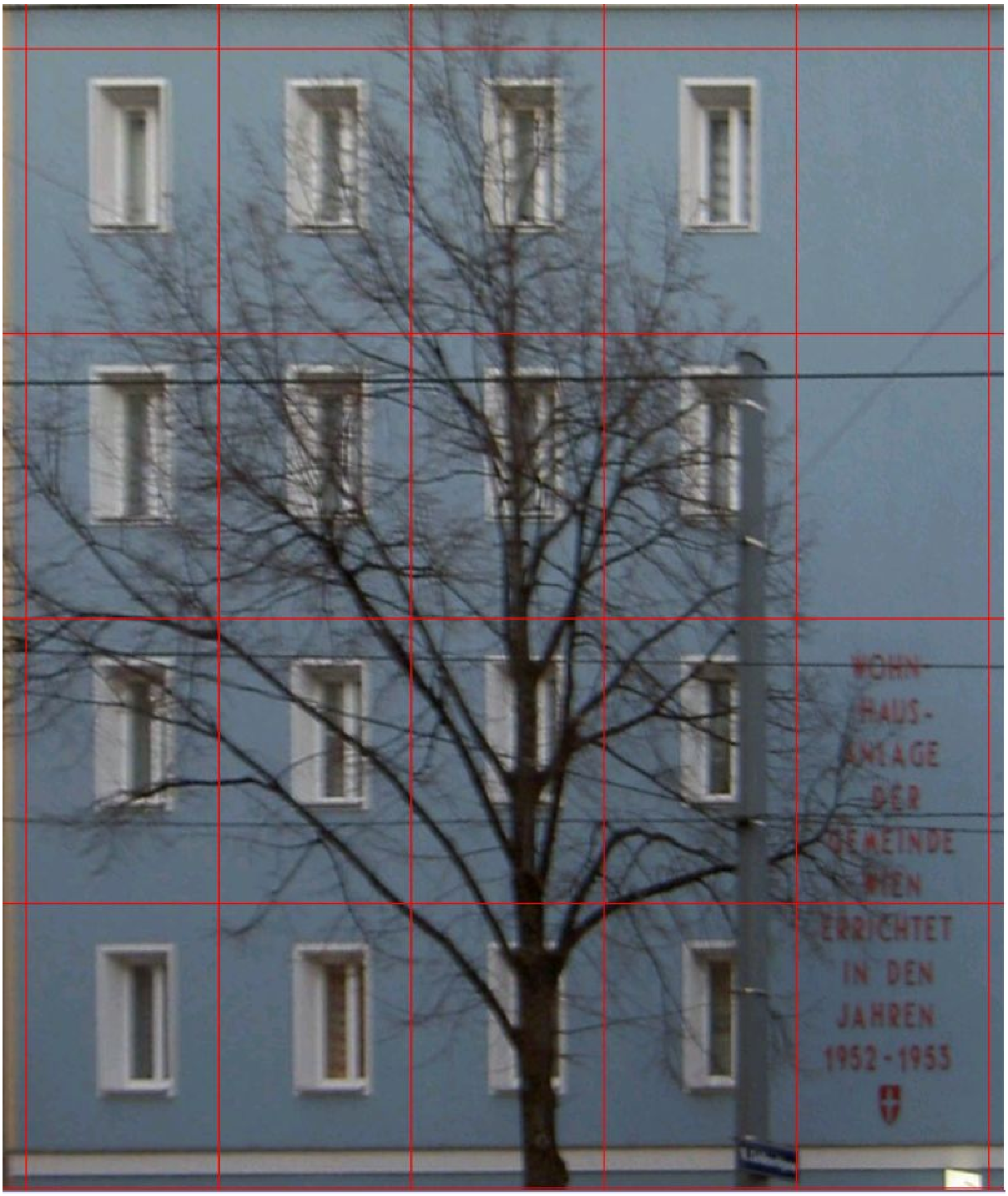
Obviously the two different sampling methods behave completely different with random noise applied to the input images. The best-match sampling criterion is extremely robust and is even under heavy interferance with random noise able to find the regular pattern beneath. Threshold sampling, on the other hand, is quite fragile with noisy images. This is due to the fixed similarity threshold criterion which must be fullfilled 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.

Robustenss against real-world artifacts. The previous tests demonstrate the method's robustness using artificial measurable image damage. While such artificial settings are good for quantifying the degree of robustness of the algorithm for objective comparison against other methods they do not give true evidence of the robustness under real-world circumstances with common artifacts and interferences on ground-based fotographed facade images.

The method presented herein has been tested against many images with common urban artifacts disturbing the test facade images such as:

- Shadows of building silhouettes
- Hard reflections of sunlit windows
- Traffic signs and lights, cables, signs and advertisement
- Vegetation
- Blur from moving camera
- Transistor noise on underexposed shots

The following image should demonstrate the robustness of the segmentation algorithm on a facade image that is both unsharp and heavily obscured by trees and other typical objects of a urban environment.

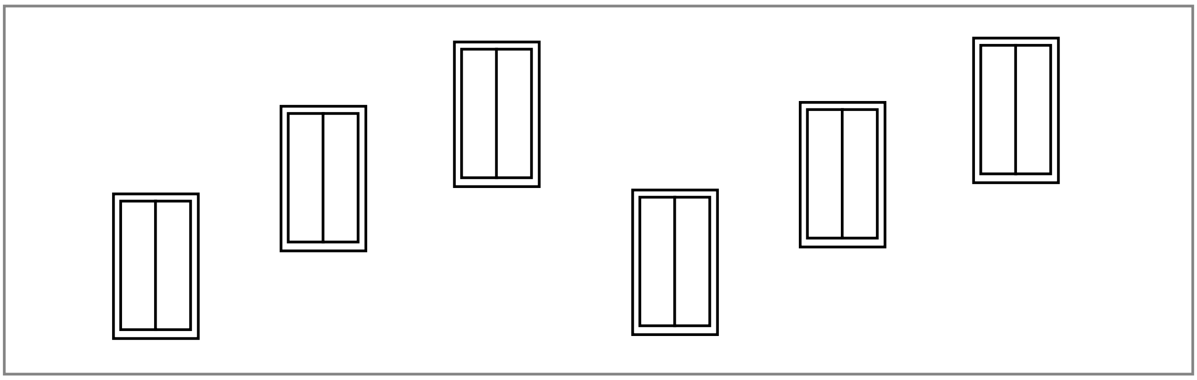


This segmentation has been calculated by drawing one hundred random samples using the best-match sampling method. You can see that the algorithm reliably detects the repetitive pattern even though it is heavily obscured by blur and irregular vegetation.

3.3. Limitations

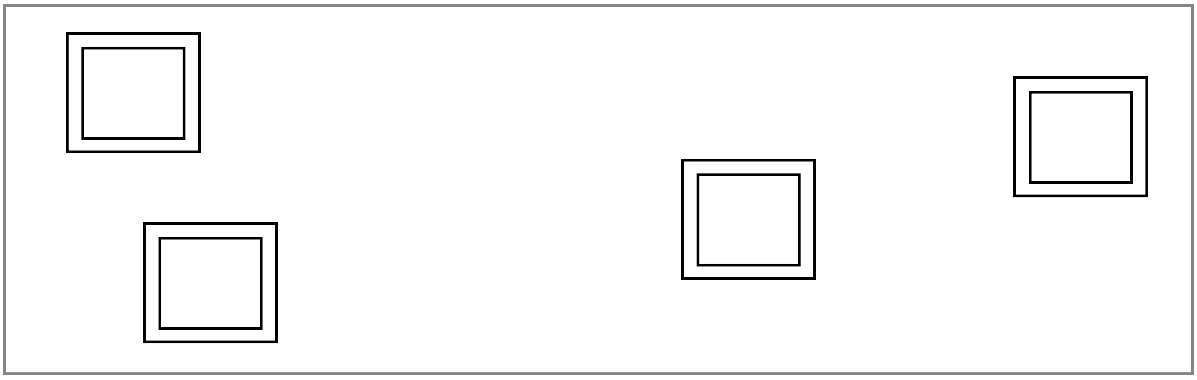
Although the presented methods have been shown to be quite robust and reliable on typical facades they also have their limitations. Due to their non-hierarchic ortho-grid-based nature the algorithms cannot detect patterns that are non-orthogonal or hierarchically structured. In the following paragraphs the most important limitations are summarized and example images are given.

Non-orthogonal grid-aligned features.



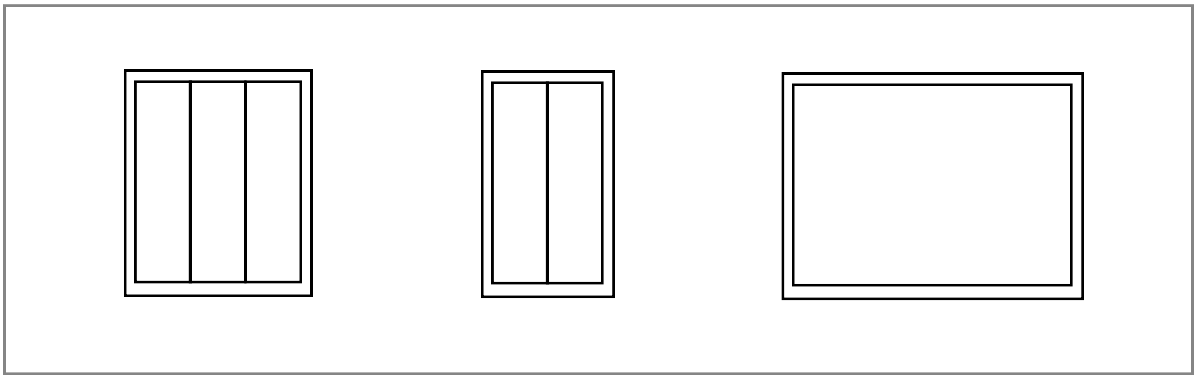
The algorithm's search strategies are intelligent and efficient but they search only in horizontal and vertical directions. Facade features that are not aligned orthogonally are therefore not supported even though they may be aligned on a non-orthogonal grid. In this example the method would split after the third window because there are two orthogonally aligned instances of a group of three windows.

Randomly placed features.



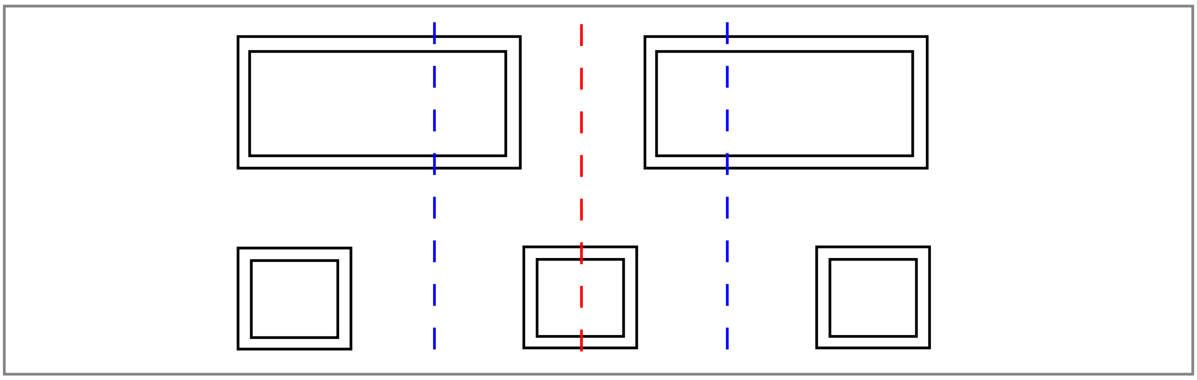
Facade features that are not structured on a grid and not aligned orthogonally are not supported by the presented method. In this example the method would not find any split lines at all.

Regularly aligned but non-uniform sized features.



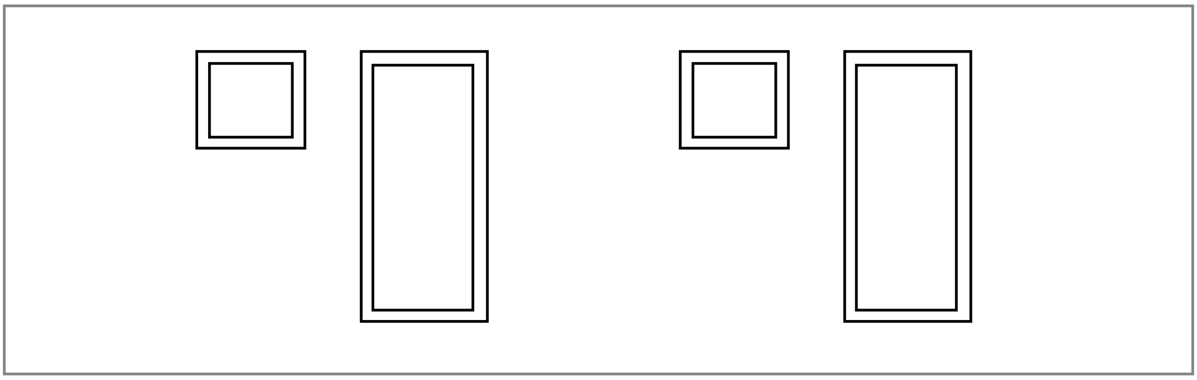
Facades sometimes feature different types of windows in different sizes. Even if these differently sized windows are orthogonally aligned on a grid the segmentation might return incorrect results, depending on the degree of difference between the different window styles.

Other than translational symmetries.



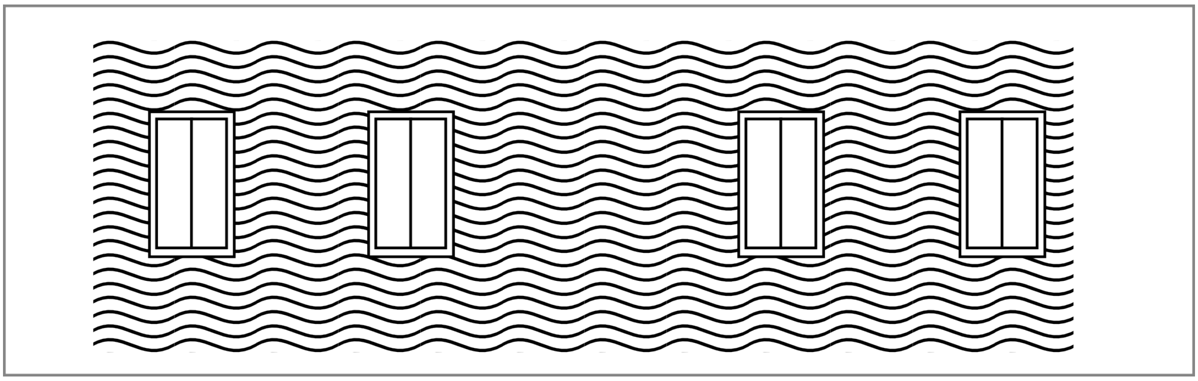
The method is based on efficient detection of translational symmetries. If features are aligned, for example, in mirror symmetry the segmentation will return incorrect results. In fact, the algorithm could be modified to also support other symmetries. This is a topic for future work.

Alternating translational symmetry or hierarchic grouping.



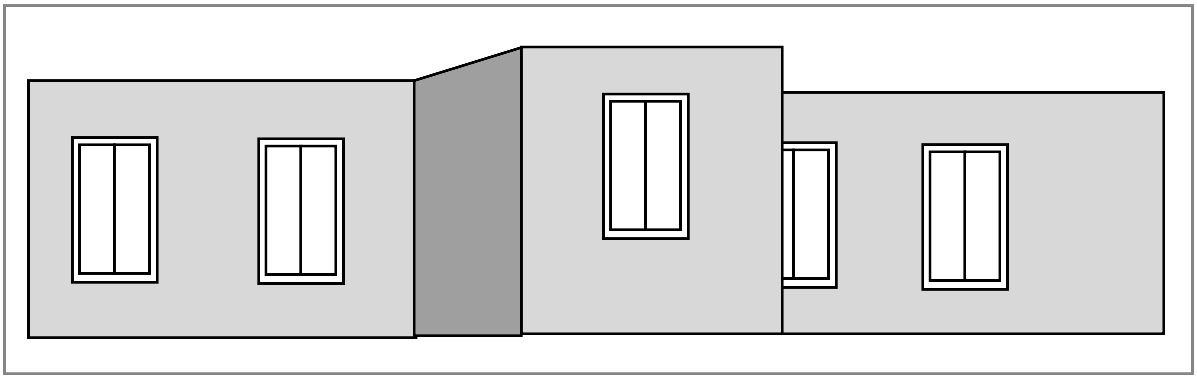
Alternating patterns are a special case of overlapping patterns which can not currently be correctly handled by the method. Also more complicated hierarchical structures such as recursive grouping of patterns are not supported. It is not easy or probably impossible to extend these kind of search algorithms to detect recurisve hierarchic patterns.

Spatially overlapping patterns.



Overlapping patterns can possibly cause wrong segmentations when the pattern that is actually only decorative background is visually stronger, features more edges and measures higher similarity values. In this case, the image would be correctly segmented because the high-frequent pattern is filtered out in the higher pyramid levels of the multi-resolution similarity sampler. Nevertheless, when two patterns are overlapping, only one is detected and the segmentation may not give the desired result.

Regular patterns that are placed on different facade planes.



Due to perspective distortions of all geometry that is off the chosen orthogonalization plane during the rectification stage, where a perspectively distorted facade is warped into a rectilinear image, facade patterns which could otherwise be sufficiently detected and segmented may not be processable depending on the degree of distorion. This is a special case and/or combination of some of the previously discussed limitations.

Chapter 4.

Conclusions

4.1. Summary

The problem of building facade segmentation is very complex and expensive. The problem is so difficult because the appearance and arrangement of windows in urban environments varies widely, such that it is impossible to successfully apply typical image processing approaches such as model fitting or pixel-based segmentation methods for non-specific urban facade images. Most of the methods that currently exist either require manual input or are designed for a very specific set of facades. Currently, there are no generic fully automatic approaches that work for a large number of different facades.

Striving for an automatic and totally appearance independent detection and segmentation algorithm, this thesis is based on the simple idea of exploiting the inherently repetitive nature of facade elements of typical buildings. The thesis presents methods to efficiently detect the inherently regular repetitive patterns of typical urbane facades and to to segment such images into facade tiles for further processing such as geometry extraction.

Starting from orthogonalized photographs of building facades the first step to solve the problem is the search for repetitive patterns in horizontal and vertical directions. Of course, the number of possible patterns their offset of repetition, position and spatial extent is known to be very large. In other words, the search space is high dimensional. To search this problem space for all possible solutions is not efficient and would not be feasible with respect to computation time, especially for large input images. The efficient solution of the search problem might be this thesis' most significant scientific contribution in the field of urban reconstruction: a Monte-Carlo based image sampling algorithm which reduces the computational complexity by a large degree (compared to exhaustive search) albeit the high dimensionality of the problem space. Using intelligent importance sampling strategies the performance of the Monte-Carlo approach is further improved.

The two sampling strategies presented in section 2.2.8, *best match criterion* and *threshold criterion* provide different trade-offs between quality and efficiency. Basically the sampling algorithm can be explained like this; pick any location in the image at random (a sample). When using the *threshold criterion*, calculate the similarity to a point that is offset horizontally or vertically by a random distance. If the similarity is higher than a certain threshold (i.e. 0.8 in a range of 0.0 to 1.0 where 1.0 means equal and 0.0 means totally different) the offset is stored in a histogram for later analysis. When using the *best match criterion* check all points horizontally or vertically for their similarity to the initially chosen random point. Select the point with the highest similarity and store the offset in the histogram. Take N more random samples and repeat the sampling process described above for each of them. The number of samples required for similar results is different for the two sampling strategies but both have their advantages. See the results in chapter 3 for a detailed comparison of the two.

Before we talk about the analysis of the gathered sampling results, I must say a few words about the similarity measure on which the sampling process is based on. The method of choice is *Normalized Cross-Correlation* of small image patches, which can be computed very efficiently and provides a good measure. This approach for measuring the similarity of image locations is state of the art in image processing and computer vision. However, for our application simple Normalized Cross-Correlation was not robust enough. Therefor, the similarity operator was improved by sampling in multiple image resolutions (i.e. on all layers in an image pyramid). Section 2.2.5 describes this generalized multi-resolution image space similarity operator which significantly boosts robustness and quality of the sampling stage. Even better, sampling the multi-resolution image pyramid practically makes the sampling process independent of the image size because the sampling operator can automatically adapt to the size of the sampled image features.

Next, let us take a closer look at the analysis of the sampled data. The taken samples have been sorted into a histogram of offsets. That means, in other words, we are simply counting how often a specific distance of similar points has appeared during the random sampling stage. Distances which have been recorded much more often than others are a strong hint to a repetitive image pattern that repeats it’s instances with more or less exactly that offset. Technically the extraction of such dominant offsets is done by extracting the dominant peaks in the histogram. The peak identification is implemented by *Mean-Shift* clustering. These extracted major offsets, we assume, correspond to relevant patterns in the image.

We do not care about the location of the sampled offsets at all. The problem with that approach is that we now only know that there must be relevant patterns in the image and we know their exact offset. However, we do not know their location and extent in the image area. There might be better approaches to solve that problem based on the sample data we already gathered but in this thesis it is solved by doing another random sampling stage. It can be explained like this: take each of the major offsets (which we already know that they must correspond to a dominant image pattern) and take test samples with that offset at random all over the image. We do that for each of the extracted offsets. A test sample simply measures the similarity of a random location to a point located at a distance of the offset we are testing. If the similarity is low we can assume that the pattern with that offset is not located here, if it is higher than the similarity of other offsets we can assume that this pattern is the most dominant at this location.

Based on the samples of the second sampling stage we can calculate a segmentation of the image into facade detail tiles, such that on every tile there is one instance of a repetitive facade element (generally speaking an instance of a repetitive image pattern). The segmentation algorithm presented in section 2.3.2 is quite simple. It is inspired by Monte-Carlo integration and works like this: we start at the border of the image and set our anchor. For each offset we add up the similarity of all samples in the image strip with a width of offset pixels. This sum of sampled similarities is weighted by the area of the image strip. We take the offset with the highest weighted sum of similarity samples and move the anchor by this offset. A splitting line is set. Then we repeat the integration and setting of anchor and splitting line until we reach the other border of the image. When you do this for both directions (horizontal and vertical) you get a grid that segments the locally dominant repetitive image pattern instances.

The idea of such an automatic segmentation method is to be able to base facade detail reconstruction methods on it. It solves the difficult and expensive search problem quite efficiently and, as the results show, is also very robust. We go into more detail about possible further work in section 4.4.

4.2. Implementation

The reference implementation for the WikiVienna project accompanying this thesis was conducted using the language C# version 3.5. The algorithms were integrated into the VRVis' internal computer vision and visualization toolkit. In the following sections a few interesting details about the implementation are presented.

4.2.1. Image Processing

For the feature selection and the similarity operator a highly efficient implementation of image processing routines was needed. As of time of this writing, there are no native C# image processing libraries available. Therefore the two popular C++ image processing libraries *Intel Performance Primitives* and *OpenCV* had to be prepared for use with C#.

OpenCV is a computer vision library originally developed by Intel. It is free for commercial and research use under the open source BSD license. The library is cross-platform, and runs on Windows, Mac OS X, Linux, PSP, VCRT (Real-Time OS on Smart camera) and other embedded devices. It focuses mainly on real-time image processing.

Intel's Integrated Performance Primitives (Intel IPP) is a threaded library of functions for multimedia and data processing applications, produced by Intel. The library supports Intel and compatible processors and is available for Windows, Linux and Mac OS X operating systems.

If OpenCV finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself.

We wrote managed wrapper assemblies that expose all the major methods of Intel IPP and OpenCV for use with managed .NET languages. We chose IPP and OpenCV because they are both implemented most efficiently on Intel processor architectures by parallelizing calculations as much as possible on single and multi-core processors heavily utilizing *Single Instruction Multiple Data (SIMD)* extensions of recent CPUs.

In the following paragraphs some of the image processing problems that were solved using external library functions are summarized.

Calculating an Edge Image. The edge image is a very important ingredient to the Monte-Carlo sampler described in section 2.2.6 because it allows *Importance Sampling* as described in section 2.2.7. The presented importance sampling strategy increases the performance of the Monte-Carlo sampling process (especially the best-match sampling method) by selecting only important features from the image. In our case, edges and corners have the most influence on the similarity of two image patches. Also, sampling directly on edges allows the best possible precision in terms of results.

The edge image is obtained by combining the IPP Image Processing functions *ippiFilterScharrHorizBorder, ippiFilterScharrVertBorder* and *ippiCanny* with appropriate image format parameters.

The performance of the Monte-Carlo sampling algorithms can be boosted by using an edge image. The trick is, to restrict the random samples to be drawn only from **interesting** image locations. Interesting image locations in terms of facade feature detections are discontinuities (i.e. borders of window elements). This way the sampling algorithm gives better results with less samples. It is called Importance Sampling which is explained in detail in section 2.2.7.

The similarity operator. The Monte-Carlo sampler is based on a similarity operator which is implemented as a special composition of *Normalized Cross-Correlation (NCC)* samples. The definition of a similarity operator is to calculate the similarity of two image points with respect to their environment (i.e a certain rectangular window around them). We tested quite a few different similarity measures provided by IPP but in the end, the function *ippiCrossCorrValid_NormLevel* proved to be most efficient both with respect to precision and performance. When using Cross-Correlation to measure similarity between certain image points it is essential to have an odd window size (where the term window is to be understood as defined in section 2.2.4). Using *ippiCrossCorrValid_NormLevel* instead of *ippiCrossCorrFull_NormLevel* or *ippiCrossCorrSame_NormLevel* makes a major difference with respect to performance. Since we are only interested in the correlation coefficient of the pixel where both source and target image region are exactly overlapping each other, *ippiCrossCorrValid_NormLevel* calculates only that one nothing else.

The multi-resolution similarity operator presented in section 2.2.5 can be optimized for higher overall sampling performance. The trick is to predict the similarity value with a cheap operation to the complete image pyramid if one sample already tells us that the similarity is expected to be low. In order to get a good prediction you should take that first sample from the highest possible pyramid layer which covers the largest (down-sampled) image area of the original image. If this predictive similarity value is lower than a given threshold (i.e a correlation value of <= 0.3) we can save valuable computation effort by instantly returning that value without sampling and weighting all the other image-pyramid layers.

4.3. Conclusions

The work for this thesis was 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[1]. 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 content in the image or distinguish it from uninteresting background noise.

For example, the tile segmentation is completely unaware of the underlying signal. Instead it uses the measured information on the offset of repetitive instances of a pattern to set splitting lines. Anyway, from the signal itself, without a priori knowledge, it is not possible to decide where a window begins and where it ends. Also, because of the vast variety of window types, decoration elements and colorings it is almost impossible to distinguish between facade background and facade elements in a generic way. Thus, the proposed method has a huge advantage over all content based methods: it works for any underlying signal, only if it is repetitive and orthogonally aligned.

Actually the algorithms can easily be reused for other applications that are not connected to facade analysis at all. It could be generalized for any type of pattern with any type of symmetry in any number of dimensions. The next section is going into detail about that.

The most important conclusion is that the initial idea to solely rely on translational symmetries lead to impressive robustness of the algorithms in the end. The presented segmentation method is incredibly robust with respect to obscuring noise, occlusion caused by vegetation or traffic signs and other disturbing artifacts. Such real world image defects that are common in our image material taken from Vienna would break any content-based methods (such as pixel-based segmentation, etc.) for sure.

[1] Given that the pattern instances are aligned on a rectilinear grid.

4.4. Application and Future Work

The segmentation method developed in this thesis is driven by the current development of an urban reconstruction framework at the VRVis Company, namely the WikiVienna project. It is applicable as a preprocessing step for geometric refinement and detail modeling of facade elements which is needed to improve the visual appearance of close-up views of facades. The automatic processing and high-quality geometric reconstruction of the facade tiles generated by this method is still an active research topic which has not yet been solved completely in a generic manner. At the moment, manual or semi-automatic methods are prevailing. In future work of this thesis could be extended to general search directions, to perspective images or into a recursive hierarchic method. The segmentation could be done more efficiently based on the sample data of the first sampling stage.

4.4.1. Future Work

Generalize search directions and symmetry type. In future work the sampling technique which is currently hard-coded to translational symmetries in orthogonal directions could be generalized to any direction and to support other symmetries such as mirror symmetry. This would open completely new applications for the algorithm such as texture segmentation.

Detection of recursive hierarchic symmetry groups. This would allow segmentation of almost any facade architecture and type that contains repeating similar elements. To do that the algorithm must be able to search locally not image global as it is implemented now. A procedural model based on a grammar would need to be populated and verified basing on the pattern detection algorithm.

Search in perspective images. By doing a hough transformation [Vui94] of the edges found in perspective images and extracting the significant parameters one can extract parallel lines in perspective images efficiently. The search algorithm could be extended to search in perspectively distorted facades, which would also have the positive side effect that facades of neighboring houses can be segmented easily if they do not share the same features.

Optimizing the segmentation stage. Currently the segmentation stage is not reusing the sample data from the first sampling stage, which means, the locations of the random samples are completely discarded. The segmentation stage is recalculating this information by doing a second random sampling stage. This second sampling stage might be avoided if the location information of the first stage is saved and reused. The performance of the overall method could be cut by roughly 50% or even more. Optimization was not the main focus of this thesis so this was not implemented.

Texture homogenization. Certain image artifacts of facade images taken from a real urban environment contain many unwanted debris such as distorted traffic signs, people, cars, shadows and vegetation. In order to use the images as textures for a virtual rendering of a city model automatic elimination of such artifacts is desired. The pattern detection method of this thesis could be used to homogenize the repetitive parts of the texture.

Identification of repetitive elements in 3D models. The algorithms presented herein could be extended to the third dimension in order to identify repetitive or otherwise symmetrical elements in 3D models such as building models that are created by a triangulation scanner. An implementation for 3D models would highly profit from the Monte-carlo approach which is even more efficient[2] the higher the dimensionality of the search space.

[2] Note: *more efficient* is to be understood with respect to other approaches, not in relation to lower dimensional problems.

Appendix A.

Rectification using a 2D Homography

As this method requires orthogonalized images as input the following paragraphs explain the mathematical background of rectification.

Photos taken by a standard consumer camera usually distort the depicted geometry by the camera parameters and the perspective projection. For simplicity, we will assume that the camera was calibrated and the images taken have been undistorted. This way we can easily rectify images using a simple projection between arbitrary planes called homography.

A.1. Homography

Homography is a concept in projective geometry. A homography is an invertible transformation from one projective plane to another which is characterized by mapping straight lines to straight lines. Homography is also termed *collineation*, *linear projective transformation* or *projectivity* in the literature.

Any two images of the same planar surface (i.e. a flat building facade) are related by a homography.

Given a point $\mathbf{p}_a$ on surface a and a corresponding point $\mathbf{p}_b$ on surface b and a homography matrix $\mathbf{H}$ which represents a bijective projection between the planes a and b:

$$\mathbf{p}_a = \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix}, \mathbf{p}'_b = \begin{bmatrix} w'x_b \\ w'y_b \\ w' \end{bmatrix}, \mathbf{H}_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{A.1}$$

then either point on one of the surfaces can be expressed as the matrix product of the homography matrix $\mathbf{H}$ and its corresponding projected point on the other surface:

$$\mathbf{p}'_b = \mathbf{H}_{ab}\mathbf{p}_a \tag{A.2}$$

An important property of this transformation is its bijectivity. It means the projection can be reversed by the inverse homography matrix.

$$\mathbf{H}_{ba} = \mathbf{H}_{ab}^{-1} \tag{A.3}$$

Note, that matrix multiplication can not directly express a division. This is why the homography can only be described as matrix operation in projective geometry where the points are represented as homogeneous coordinates. The result of the matrix multiplication in equation A.2 p'_b in general consists of a homogeneous component other than 1. In the mathematical concept of projective geometry p equals p'_b. However, if we want values that are equal to euclidean 2D coordinates we just need to divide through the homogeneous component w of the vector and ignore the third coordinate which is 1.

$$\mathbf{p}_b = \mathbf{p}'_b / w' = \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} \tag{A.4}$$

Using the homography we can project the perspective image into a corresponding orthogonalized image. In computer vision this process is called *rectification*.

A.2. Calculating the Homography from Corresponding Image Points

Provided that we know at least four corresponding pairs of points $\mathbf{a}_i$ and $\mathbf{b}_i = (x_i, y_i, z_i)$ in the images a and b, we can calculate the homography matrix relating the linear transformation from plane a to plane b by means of solving the resulting linear equation. First we separate the homography matrix $\mathbf{H}$ into its three basis vectors $\mathbf{P}_i$:

$$\mathbf{H} = \begin{bmatrix} P_1^{\mathrm{T}} \\ P_2^{\mathrm{T}} \\ P_3^{\mathrm{T}} \end{bmatrix} \tag{A.5}$$

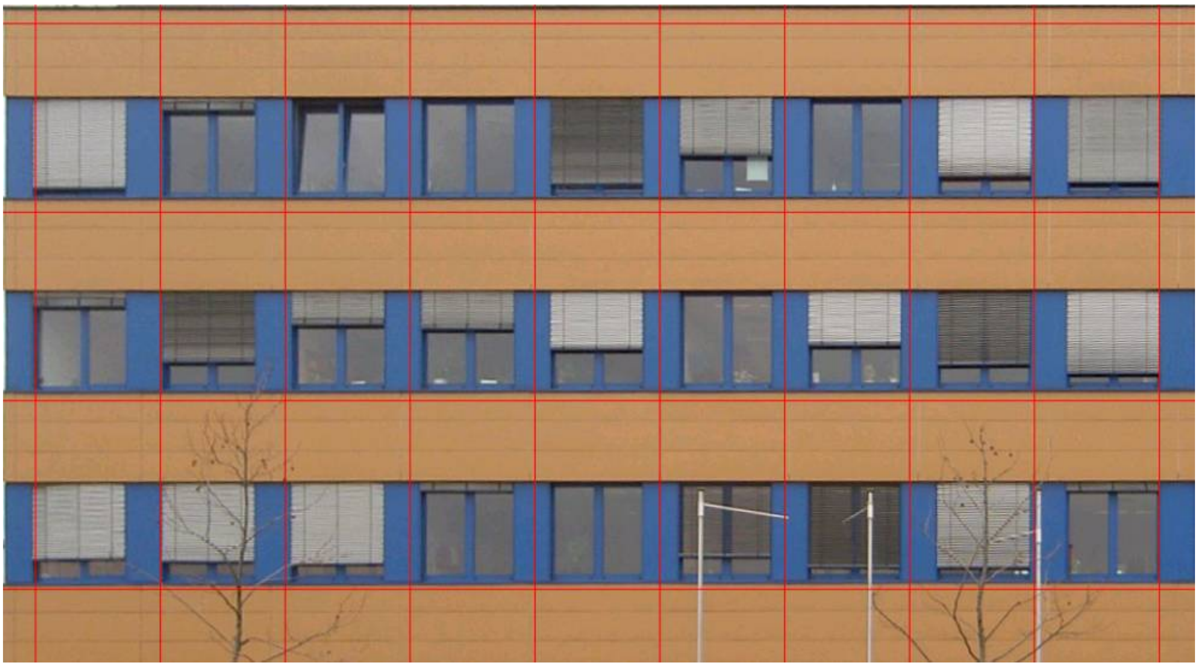
From the four corresponding pairs of points we obtain eight equations such that:

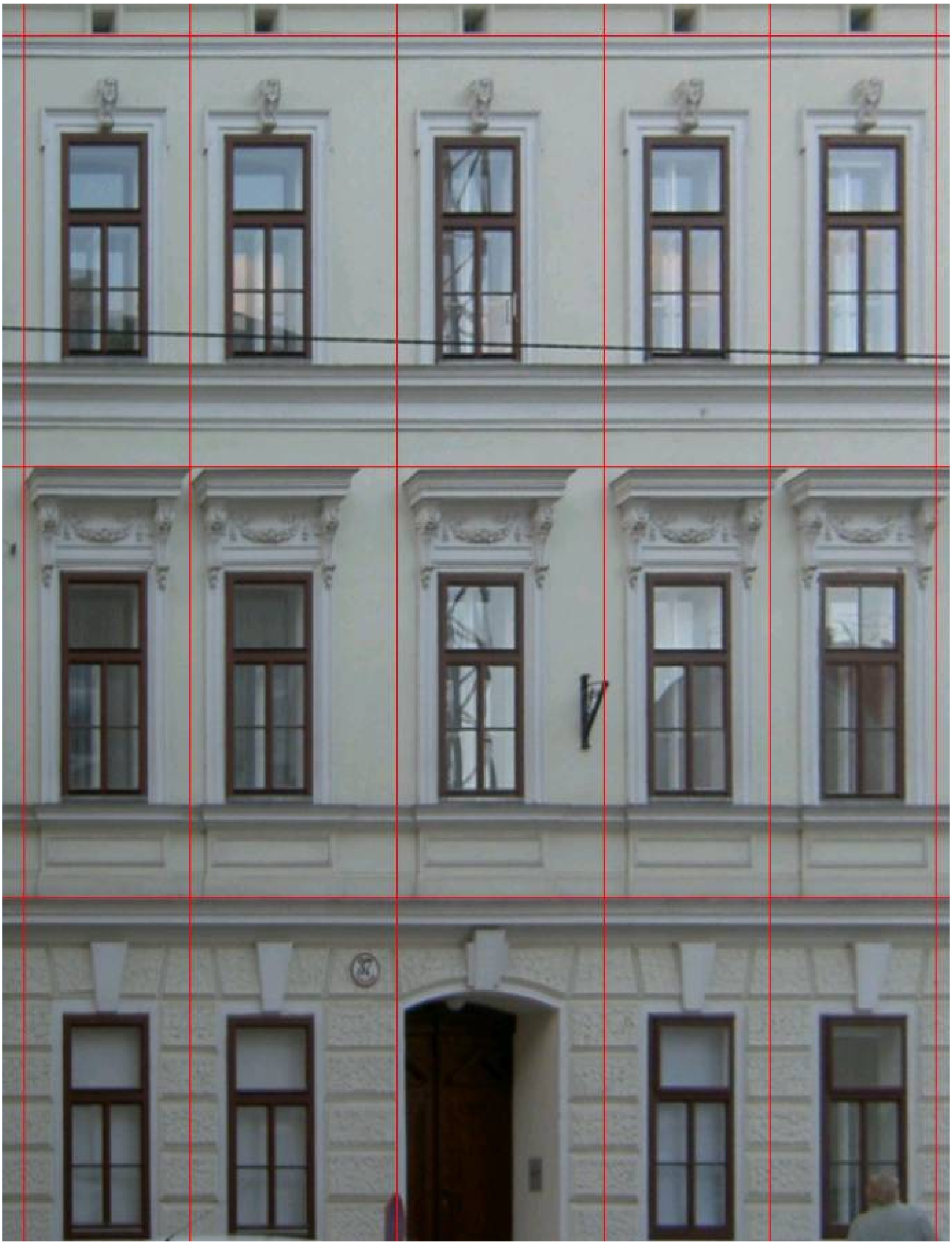
$$x_i\mathbf{P}_3\mathbf{a}_i - z_i\mathbf{P}_1\mathbf{a}_i = 0, y_i\mathbf{P}_3\mathbf{a}_i - z_i\mathbf{P}_2\mathbf{a}_i = 0 \tag{A.6}$$

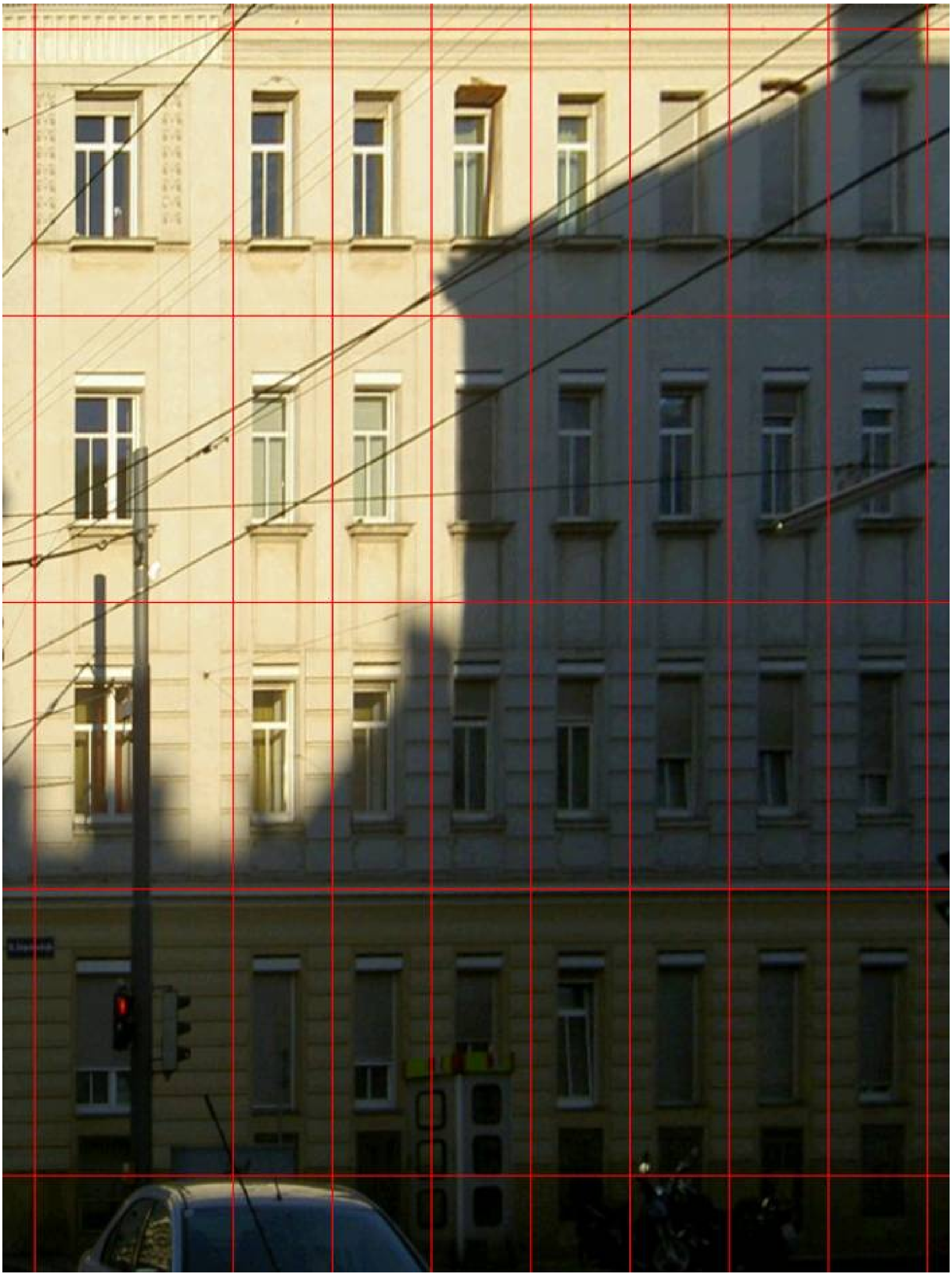
By solving this linear equation system the components of the homography matrix can be calculated.

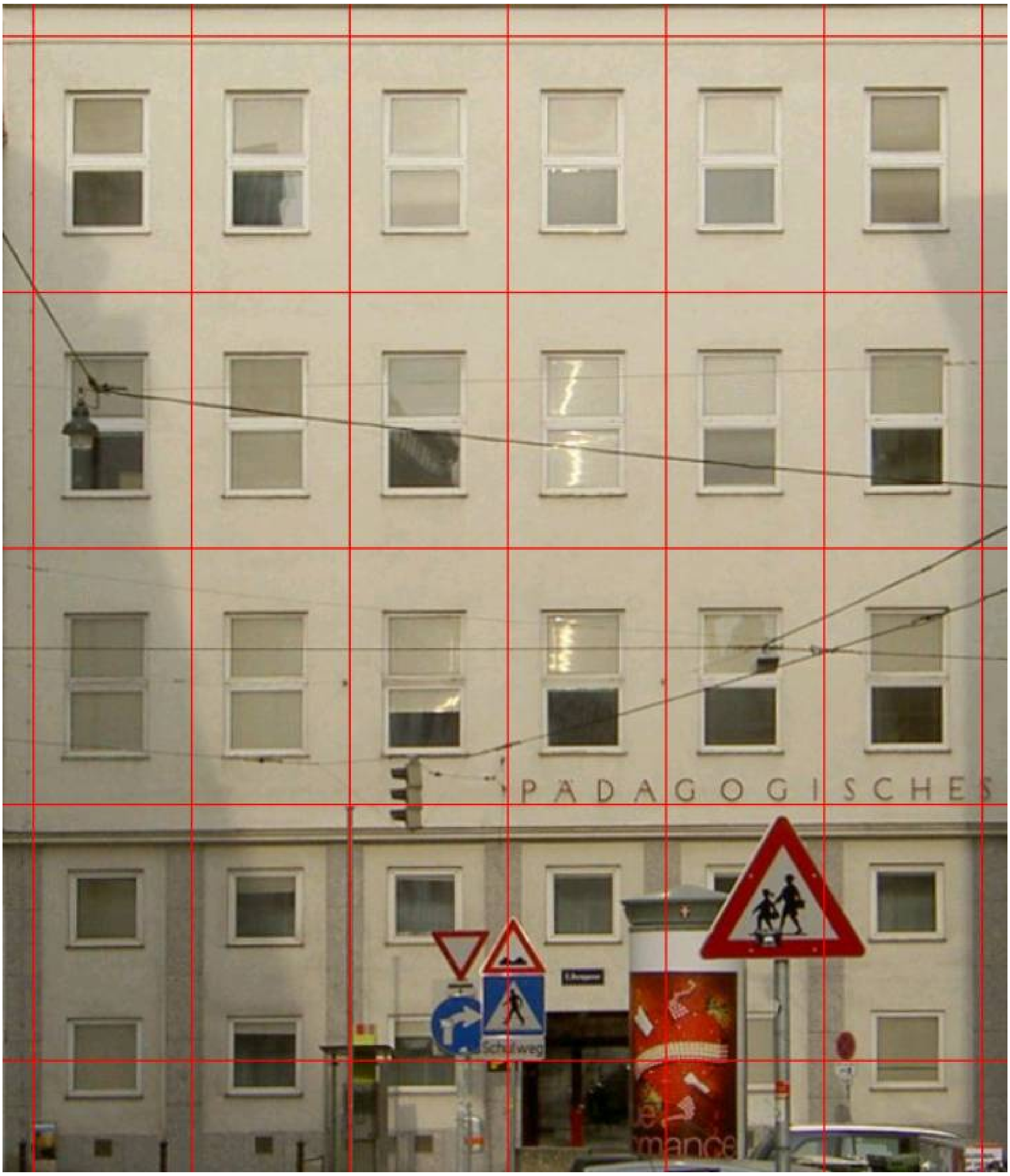
Appendix B.

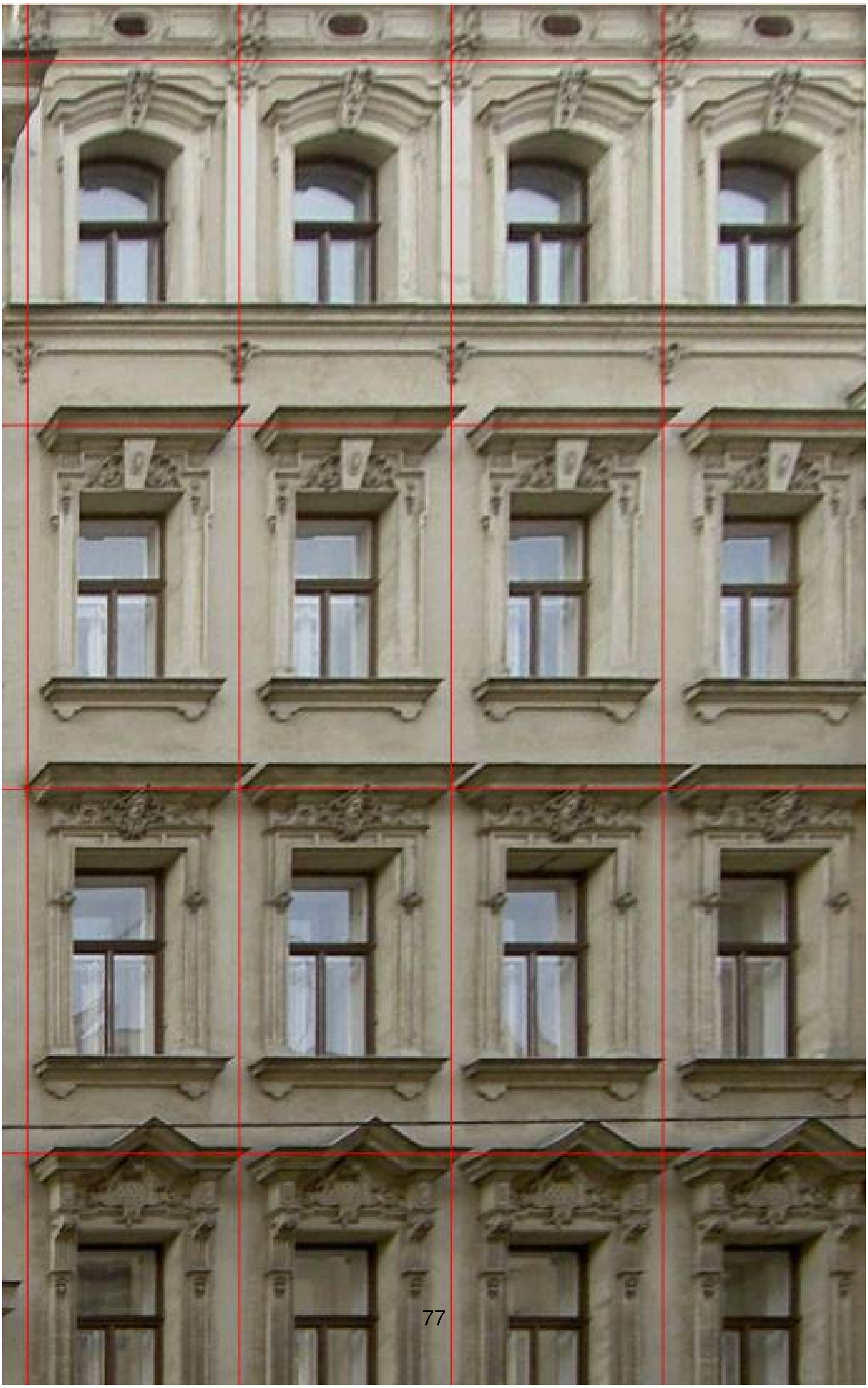
Images








PÄDAGOGISCHES
Schulweg



Bibliography

[AD04] Fernando Alegre and Frank Dellaert. A probabilistic approach to the semantic interpretation of building facades - georgia tech's institutional repository. 2004.

[ASJ+07] H. Ali, C. Seifert, N. Jindal, L. Paletta, and G. Paar. Window detection in facades. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 837–842, 2007.

[Bai97] D. G. Bailey. Detecting regular patterns using frequency domain self-filtering. In *ICIP '97: Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 1*, Washington, DC, USA, 1997. IEEE Computer Society.

[BI07] Oren Boiman and Michal Irani. Detecting irregularities in images and in video. *International Journal of Computer Vision*, 74(1):17–31, August 2007.

[BR06] C. Brenner and N. Ripperda. Extraction of facades using rjmcmc and constraint equations. In *PCV06*, 2006.

[Can86] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[CM02] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[CS07] Jan Cech and Radim Sara. Windowpane detection based on maximum aposteriori probability labeling. Technical report, 2007.

[DTC04] A. R. Dick, P. H. S. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *Int. J. Comput. Vision*, 60(2):111–134, 2004.

[HLEL06] James Hays, Marius Leordeanu, Alexei Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. pages 522–535. 2006.

[HLL01] J. T. Hsu, Li-Chang Liu, and C. C. Li. Determination of structure component in image texture using wavelet analysis. In *Image Processing, 2001.*, volume 3, pages 166–169 vol.3, 2001.

[HZ05] F. Han and S. C. Zhu. Bottom-up/top-down image parsing with attribute grammar. Technical report, January 2005.

[LCT04] Yanxi Liu, Robert Collins, and Yanghai Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE PAMI 2004*, 26(3):354 – 371, March 2004.

[LN04] S-C Lee and R. Nevatia. Extraction and integration of window in a 3d building model from ground view images. In *Computer Vision and Pattern Recognition*, 2004.

[Low03] David G. Lowe. Distinctive image features from scale-invariant keypoints. 2003.

[MR07] H. Mayer and S. Reznik. Building facade interpretation from uncalibrated wide-baseline image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(6):371–380, February 2007.

[MZWVG07] Pascal Müller, Gang Zeng, Peter Wonka, and Luc Van Gool. Image-based procedural modeling of facades. *ACM Trans. Graph.*, 26(3), July 2007.

[SB03] Konrad Schindler and Joachim Bauer. A model-based method for building reconstruction. In *HLK '03: Proceedings of the First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, Washington, DC, USA, 2003. IEEE Computer Society.

[SI07] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[TS08] Thorsten Thormählen and Hans P. Seidel. 3d-modeling by ortho-image generation from image sequences. In *ACM SIGGRAPH 2008*, pages 1–5, New York, NY, USA, 2008. ACM.

[Tsa06] Liu J-K Hsiao K-H Tsai, F. Morphological processing of video for 3d building model visualization. In *Proc. 27th Asian Conference on Remote Sensing (ACRS2006)*, Ulaanbaatar, Mongolia., 2006.

[TTVG01] Andreas Turina, Tinne Tuytelaars, and Luc Van Gool. Efficient grouping under perspective skew, 2001.

[VG07] Zeng-G. Van den Borre F. Müller P. Van Gool, L. Invited paper: Towards mass-produced building models. In *PIA07 - Photogrammetric Image Analysis*, 2007.

[Vui94] Jean E. Vuillemin. Fast linear hough transform. In *International Conference on Application-Specific Array Processors*, pages 1–9. IEEE Computer Society Press, 1994.

[WTT+02] Xiaoguang Wang, Stefano Totaro, Franck Taill, Allen R. Hanson, and Seth Teller. Recovering facade texture and microstructure from real-world images. In *In Proc. 2 nd International Workshop on Texture Analysis and Synthesis at ECCV*, pages 381–386, 2002.

[XFT+08] Jianxiong Xiao, Tian Fang, Ping Tan, Peng Zhao, Eyal Ofek, and Long Quan. Image-based façade modeling. *ACM Trans. Graph.*, 27(5):1–10, 2008.

[ZG02] Steve Zelinka and Michael Garland. Towards real-time texture synthesis with the jump map. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 99–104. Eurographics Association, 2002.