
TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DISSERTATION

# Processing of Façade Imagery

ausgeführt
zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften

unter der Leitung von
**Univ.-Prof. Dipl.-Ing. Dr.techn. Werner Purgathofer**
Institut für Computergraphik und Algorithmen (E186)
Technische Universität Wien

und unter Mitwirkung von
**Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Peter Wonka**
Department of Computer Science and Engineering
Arizona State University

eingereicht an der
Technischen Universität Wien
Fakultät für Informatik

von
**Dipl.-Mediensystemwiss. Przemyslaw Musialski**
Matrikelnummer: 0727665
Pichelwangergasse 42
1210 Vienna

Wien, am 19. Oktober 2010

**Przemyslaw Musialski**

# Processing of Façade Imagery

Vienna · October 2010

# Abstract

Modeling and reconstruction of urban environments is currently the subject of intensive research. There is a wide range of possible applications, including virtual environments like cyber-tourism, computer games, and the entertainment industries in general, as well as urban planning and architecture, security planning and training, traffic simulation, driving guidance and telecommunications, to name but a few. The research directions are spread across the disciplines of computer vision, computer graphics, image processing, photogrammetry and remote sensing, as well as architecture and the geosciences. Reconstruction is a complex problem and requires an entire pipeline of different tasks.

In this thesis we focus on processing of images of façades which is one specific subarea of urban reconstruction. The goal of our research is to provide novel algorithmic solutions for problems in façade imagery processing. In particular, the contribution of this thesis is the following:

First, we introduce a system for generation of approximate orthogonal façade images. The method is a combination of automatic and interactive tools in order to provide a convenient way to generate high-quality results.

The second problem addressed in this thesis is façade image segmentation. In particular, usually by segmentation we mean the subdivision of the façade into windows and other architectural elements. We address this topic with two different algorithms for detection of grids over the façade image.

Finally, we introduce one more façade processing algorithm, this time with the goal to improve the quality of the façade appearance. The algorithm propagates visual information across the image in order to remove potential obstacles and occluding objects.

The output is intended as source for textures in urban reconstruction projects. The construction of large three-dimensional urban environments itself is beyond the scope of this thesis. However, we propose a suite of tools together with mathematical foundations that contribute to the state-of-the-art and provide helpful building blocks important for large scale urban reconstruction projects.

# Kurzfassung

Modellierung und Rekonstruktion von städtischen Gebieten ist derzeit Gegenstand intensiver Forschung. Der Hauptgrund dieser Anstrengung ist das breite Spektrum möglicher Anwendungen von detaillierten Computermodellen. Beispiele davon sind virtuelle Umgebungen für Cyber-Tourismus, Computerspiele, und die Unterhaltungsindustrie im Allgemeinen, sowie Stadtplanung und Architektur, Sicherheitsplanung und Ausbildung, Verkehrssimulation, Navigation und Telekommunikation. Auch die Forschungsrichtungen sind über verschiedene Gebiete gestreut, wie Computer Vision, Computergrafik, Bildverarbeitung, Photogrammmetrie und Fernerkundung sowie Architektur und Geowissenschaften. Rekonstruktion ist ein komplexes Problem und es erfordert eine ganze Palette von verschiedenen Aufgaben.

In dieser Dissertation konzentrieren wir uns auf die Verarbeitung von Bildern von Fassaden und damit auf einen bestimmten Teilbereich der maschinellen Stadtrekonstruktion. Das Ziel unserer Forschung ist es neue algorithmische Lösungen für spezielle Probleme in Fassadebildverarbeitung zu entwickeln. Der Beitrag dieser Arbeit ist der folgende:

Zuerst stellen wir ein System zur Erzeugung von annähernd orthogonalen Ansichten von Fassaden vor. Diese Methode ist eine Kombination aus automatischen und interaktiven Werkzeugen und bietet eine bequeme Möglichkeit qualitativ hochwertige Ergebnisse zu generieren.

Das zweite Problem, welches in dieser Arbeit behandelt wird, ist Fassadensegmentierung. In der Regel unter Segmentierung versteht man die Unterteilung der Fassade in Fenster und andere architektonische Elemente. Wir sprechen dieses Thema mit zwei verschiedenen Algorithmen zur Erkennung von regulären Strukturen in Fassadenbildern an.

Schließlich stellen wir einen Fassadenbild-Verbesserungsalgorithmus vor, mit dem Ziel, die Qualität des Fassaden-Aussehens zu verbessern. Der Algorithmus propagiert visuelle Information über das Bild, um mögliche Hindernisse und verdeckende Objekte zu entfernen.

Die Ausgabe unserer Algorithmen ist als Quelle für Texturen in Stadtrekonstruktionsprojekten bestimmt. Die Entwicklung von großen dreidimensionalen urbanen Umgebungen selbst sprengt den Rahmen dieser Arbeit. Allerdings erarbeiten wir eine Sammlung von Werkzeugen zusammen mit ihren mathematischen Grundlagen, die zum Stand der Wissenschaft beitragen und als Bausteine in komplexen Rekonstruktionsprojekten zum Tragen kommen können.

# Acknowledgments

This dissertation would not have been possible without the help of many people. First of all I would like to thank my supervisor, Werner Purgathofer, for his encouragement and for giving me the opportunity to study at TU-Vienna. Furthermore, I would like to gratefully acknowledge my co-advisor, Peter Wonka, for over two years of collaborative research over the long distance. With his advice, as well as with countless discussions and exchanges of ideas, he helped me to keep to the point and to bring this thesis finally to the conclusion.

Further, I would like to express my gratefulness to colleagues at the VRVis Research Center, who were immensely helpful over the three years of my research there. I owe thanks to Robert Tobler and to Stefan Maierhofer, who brought me to VRVis and supported me during my doctoral research. Equally, I would also like to thank all my other colleagues, most notably Andreas Reichinger, Bernd Leitner, Christian Luksch, Harald Steinlechner, Irene Reisner-Kollmann, Martin Brunnhuber, Matthias Buchetics, Michael Schwärzler, Murat Arikan, Thomas Ortner, and all others at the VRVis. A special thank is due to my students, Mike Hornacek, and especially Meinrad Recheis, for his master thesis that contributed a lot to my research.

Additionally, I would like to thank Michael Wimmer for giving me the opportunity to work at the Institute for Computer Graphics and Algorithms, and I would like to express my appreciation to Georg Stonawski of VRVis for financial support of my research, which was mainly part of the project "WikiVienna" funded by Vienna Science and Technology Fund (WWTF).

Finally, I wish to express my deep gratitude to my beloved wife, Karina. Her encouragement, faith and inspiration, as well as her efforts to keep me well gave me the strength to come through tough times and to finish this thesis, even if sometimes things did not work as they had to.

*dedicated to my parents*
*Teresa and Zbigniew*

# Contents

# 1. Introduction

Cities are the centers of our world. They are fascinating, attracting, inspiring, and even forwarding the mankind since all times. They have attracted a lot of attention in literature and science and have been investigated from many points of view in a broad spectrum of disciplines: from social over cultural, economical to technological. In this dissertation we elaborate on cities as well – we look at them from the computer science point of view.

We are interested in virtual urban worlds whose generation has become affordable in recent time due to the enormous expansion of information technology in the last twenty years. Nowadays, modeling and reconstruction of urban environments has become an estimated multi-billion dollar industry and it is currently the subject of intensive research. The reason of this effort is the wide range of possible applications, including cyber-tourism, computer games, and the entertainment industries in general, all of which have recognized the potential of virtual worlds. The generation of large urban environments has been a key aspect of several recent movies and computer games, with modeling times reaching several man years. Also, the digital maps industry has become ubiquitous and can be encountered in many everyday usage items like mobile phones and cars. Other prominent examples are the two versatile projects Google Earth and Microsoft Virtual Earth. From an economical standpoint there is an enormous benefit of being able to quickly generate high-quality digital worlds in the growing virtual consumption market.

Besides entertainment-inspired applications, city planners, local governments, and scientists are also using advanced three-dimensional simulation and visualization software to plan and manage traffic, public transportation, telecommunication, water resources, and green spaces. In addition, these stakeholders seek to better understand urban resiliency, urban sprawl, impact of large urban projects (e.g. airports), and urban development. Typically, data acquisition is a significant obstacle to using the aforementioned simulations and analyses. Furthermore, security training, emergency management, and civil protection and disaster control as well as driving simulation can benefit from virtual urban worlds.

Urban modeling and reconstruction is also very suitable for archaeological research. The challenge in archaeological modeling is to combine data acquired directly from building remains with knowledge from diverse sources, such as books, old maps, and paintings. Archaeologists can use this technology to reconstruct ancient environments by using the acquired data as constraints while encoding other sources of information in the knowledge database. Additionally, the generated models are suitable for education in architecture and urban planning.

Enabling the aforementioned goals implicitly involves a synergistic effort spanning multiple research fields, including computer graphics, image processing, computer vision, pattern recognition, photogrammetry and remote sensing, computer aided design, geosciences, and mobile-technology. Urban reconstruction is a massive and complex problem requiring a large and diverse pipeline of tasks such as data acquisition, data modeling, 3d reconstruction, geometry extraction and texture generation.

In this thesis, we seek to provide a computer graphics contribution to the interdisciplinary field of reconstruction, modeling, and user interaction. In light of the ever vanishing borders between computer graphics, image processing and computer vision, we also span multiple fields. However, we focus on methods related to image-based modeling using images as the input source.

## 1.1. Challenges

The ultimate goal of most computer based reconstruction approaches is to provide as automatic solutions as possible, but it is usually not feasible to reach fully automatic systems. The related vision problems quickly result in huge optimization tasks, where global as well as local, and top-down as well as bottom-up processes need to be considered. Especially automatic recognition tasks are affected by these problems. This means that global processes are based on local circumstances and processes, whose parameters often depend on global estimates. This problem also appears in the top down-bottom up paradox: The detection of regions of interest is both context-dependent (top down), since we expect a well defined, underlying subject, and context free (bottom-up), since we do not know the underlying subject and want to estimate the model from the data. This issue is generally known as the “chicken or egg” dilemma.

There is no unique solution to this fundamental problem of automatic systems. Most solutions try to find a balance between these constraints and are located somewhere between them or try to combine two or more passes over the data (e.g., top-down and bottom-up [HZ09]).

Often, an additional price to pay for automation is the loss of quality. From the point of view of interactive computer graphics, the quality of solutions of pure computer vision algorithms is quite low, while especially for high-quality productions like the movie industry, the expected standard of the models is very high. In such situations the remedy is either pure manual modeling or at least manual quality control over the data.

For these reasons many recent approaches employ compromise solutions that cast the problem in such a way that both the user and the machine can focus on tasks which are easy to solve for each of them. Simple user interactions, which can be performed even by unskilled users, often provide the quantum of knowledge that is needed to break out of the “chicken or egg” dilemma.

## 1.2. Problem Statement

The particular research problem which is the subject of this dissertation is the question of how to optimally process the façade image in order to obtain maximum quality textures for virtual city models. This research subject is inspired by problems that occur in practice in the reconstruction of urban environments due to the limitations given during the data acquisition process.

For example, in general it is usually not possible to obtain an approximated orthogonal projective photo of a façade. Such image could be simulated by a frontal photography taken with a telescope lens from a very wide distance. In reality, façades lie in streets which are often narrow and such a photo cannot be shot because of other buildings present in the neighborhood. Further, it is hardly possible to capture the entire façade with one camera shot. One possible remedy is either a spherical (fish-eye) or a very wide angle lens that covers a bigger viewport. Unfortunately, photos from such lenses usually suffer under strong distortions. Another possibility is to take multiple camera shots and to combine them into a single image. However, in any case post-processing is required in order to generate the final façade image.

The second common obstacle, which occurs very frequently in urban imagery, is the problem of unwanted objects in front of the buildings, such as traffic lights, street signs, vehicles, and pedestrians. In general it is not possible to recover the visual information that has been missed through such an obstruction.

In order to solve these problems, we formulate a set of research questions which split the subject into particular steps:

- How can we generate an optimal orthogonal view of a façade from a set of perspective photographs taken from the ground by a typical hand-held consumer camera?
- How can we generate an accurate, plausible tiling of an approximately orthogonal façade such that repetitive elements can be determined? Further constrains are both competitive speed and quality.
- How can we generate high-quality façade image that is free of obstacles and occluding objects from a single, approximately orthogonal façade image?

## 1.3. Contribution

The contribution of this thesis is a set of solutions which address the questions mentioned in the previous section.

In Chapter 3 we propose a method to generate an approximate orthogonal façade texture based on a set of perspective input photographs. Our approach is to sample a rectified approximation over the façade-plane from the input sources. In order to avoid kinks and seams which may remain on transitions between pixels from different source images we

introduce post-processing steps, like color adjustment and gradient domain stitching by solving a global Poisson integration. The output of this stage can serve as input for the algorithms presented in Chapters 4 and 5.

One of the still challenging tasks in façade processing is the detection and segmentation of details such as windows and ornaments. These are considered key elements of realistic representations of urban environments. In this context, the windows of typical buildings can be seen as patterns that occur multiple times within a rather regular arrangement. Considering a building's façade on a frontal and orthogonal image, the search for the dominant features can be restricted to only the axis-aligned horizontal and vertical directions.

In Chapter 4 we propose a method that handles precisely such façades and assumes that there must be horizontal and vertical repetitions of similar patterns. Using a Monte Carlo sampling approach, this method is able to segment repetitive patterns on orthogonal images along the axes even if the pattern is partially occluded. Additionally, it is very fast and can be used as a preprocessing step for finer segmentation stages. The output of this stage served usually also as input to the image enhancements algorithm presented in Chapter 6.

In Chapter 5 we introduce the second novel, data driven method to infer distributions of rectilinear grids over a simple, orthographic-rectified façade image. This approach is inspired by unsupervised learning methods like data clustering and matrix factorization. This methods allows to segment façades in a global manner and to omit any local feature detection operations.

Finally we address the problem of removing unwanted image content in a single view orthographic façade image. We exploit the regular structure present in building façades that can be detected, e.g., by the method proposed in Chapter 4. Guided by the detected symmetry prevalent in the image, we introduce a propagation process removes larger unwanted image objects such as traffic lights, street signs, or cables as well as smaller noise, such as reflections in the windows. This method is described in Chapter 6.

### Contributing Scientific Publications

During the research on the topic of this thesis, we have published the following scientific papers and reports [MWR*09, MRM*10, MLS*10, Mus10]:

- Przemyslaw Musialski, Christian Luksch, Michael Schwärzler, Matthias Buchetics, Stefan Maierhofer, and Werner Purgathofer. **Interactive Multi-View Façade Image Editing.** In Vision, Modeling, Visualisation (VMV'10), 2010.
- Przemyslaw Musialski, Meinrad Recheis, Stefan Maierhofer, Peter Wonka, and Werner Purgathofer. **Tiling of Ortho-Rectified Façade Images.** In Spring Conference on Computer Graphics (SCCG'10), Budmerice, 2010.
- Przemyslaw Musialski, Peter Wonka, Meinrad Recheis, Stefan Maierhofer, and Werner Purgathofer. **Symmetry-Based Façade Repair.** In Marcus A. Magnor,

Bodo Rosenhahn, and Holger Theisel, editors, Vision, Modeling, Visualisation (VMV'09), pages 3–10. DNB, 2009.

- Przemyslaw Musialski. **Axis-Aligned Segmentation of Orthographic Façade Images.** Technical Report Nr VRVIS-009-2010, VRVis Research Center, Vienna, June 2010.

Furthermore, during the research on the thesis, the author has contributed to a number of other publications:

- Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. **Tensor Completion for Estimating Missing Values in Visual Data.** In 2009 IEEE 12th International Conference on Computer Vision (ICCV'09), Kyoto, Japan, 2009. IEEE.
- Matthias Baldauf and Przemyslaw Musialski. **A Device-aware Spatial 3D Visualization Platform for Mobile Urban Exploration.** In The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2010), Florence, Italy, 2010. IARIA.
- Matthias Baldauf, Peter Fröhlich, and Przemyslaw Musialski. **A Lightweight 3D Visualization Approach for Mobile City Exploration.** In First International Workshop on Trends in Pervasive and Ubiquitous Geotechnology and Geoinformation GIScience conference (TIPUGG'08), 2008.
- Matthias Baldauf, Peter Fröhlich, and Przemyslaw Musialski. **Integrating User-Generated Content and Pervasive Communications - WikiVienna: Community-Based City Reconstruction.** IEEE Pervasive Computing, 7(4):58–61, October 2008.
- Przemyslaw Musialski. **Point Cloud to Model Registration.** Technical Report Nr VRVIS-009-2009, VRVis Research Center, Vienna, June 2009.

These papers are related to image processing [LMWY09], or to the topic of urban reconstruction and rendering for mobile devices as the papers by Matthias Baldauf [BFM08a, BFM08b, BM10]. Also a Technical Report about registration of point clouds to urban models has been published by the author [Mus09]. An excerpt of this report is available in Appendix B.

# 2. Related Work

This chapter is intended to provide a comprehensive overview over the work done on image-based urban modeling and reconstruction in recent years. While a considerable body of work already exists, this topic is still under very active research. The area of image-driven urban reconstruction spreads basically over three research communities: computer graphics, computer vision and last but not least photogrammetry and remote sensing. While the first two fields are clearly positioned in computer science and aim mainly at computational methods for reconstruction (CV) and interactive modeling (CG), photogrammetry developed a research strand on its own due to its roots in measuring and documenting the world. We want to point out the main concepts of this wide-spread topic.

## 2.1. Overview

It is quite a difficult task to classify all the existing approaches. In this review, we try to loosely order relevant papers from manual to automatic methods, but note that this is not always possible. In Figure 2.1, we depict the main building blocks of the reconstruction process. In this thesis, the term *modeling* is used for interactive methods, and the term *reconstruction* for automatic ones.

We omit fully manual modeling, even if it is probably still the most widely applied form of reconstruction in many architectural and engineering bureaus. From the scientific point of view, the manual modeling pipeline is generally well researched. An interesting overview of methods for the generation of polygonal 3d models from CAD-plans has been recently presented by Yin *et al.* [YWR09].

In Section 2.2, we first review methods which aim at image-based **façade processing** such as image stitching, panorama imaging or segmentation into elements such as doors, windows, and other domain-specific features. We handle the façade topic explicitly and substantially because it is of particular importance for the algorithms presented in this thesis.

In Section 2.3, we introduce automatic **sparse reconstruction** systems, often using structure-from-motion formulations. Such systems have reached a rather mature state in recent time and often serve as preprocessing stages for many other methods since they provide quite accurate camera parameters. Many methods, even the interactive ones, rely on this module as a starting point for further computations. For this reason we introduce this approach prior to any modeling or reconstruction solution.

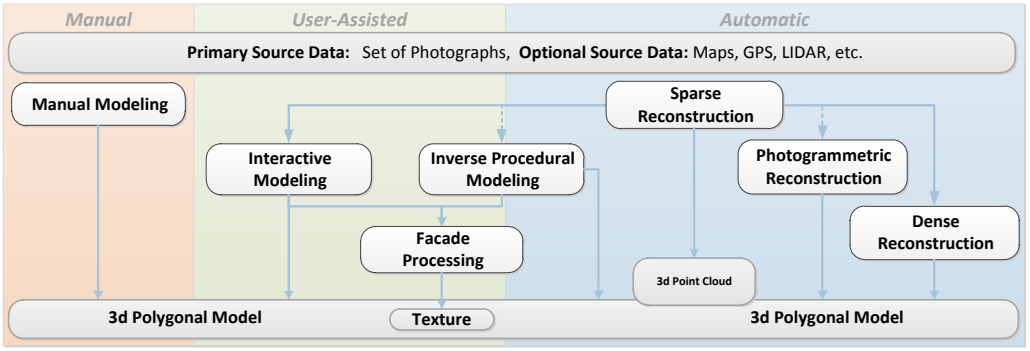


**Figure 2.1.:** *Overview of image-based reconstruction approaches, from which we omit fully manual methods like CAD-modeling. We report about interactive methods that utilize both user input and automatic algorithms as well as about fully automatic methods. Note that this is a schematic illustration and in practice many solutions cannot be classified strictly into a particular bin. Furthermore, there might be more crossover connections between particular stages.*

In Section 2.4, our survey addresses image-based **interactive modeling** approaches. Here we present a variety of concepts borrowed from fully automatic algorithms and adapted for quick interactive use.

In Section 2.5 we briefly introduce the methodology of **procedural modeling** that proved in recent time to be the tool of choice for urban synthesis. Since we are interested in urban reconstruction, we only touch this topic in order to provide a basis for the next section.

In Section 2.6, we describe the basic concepts of **inverse procedural modeling** which is recently receiving significant attention due to its ability to compute a compact and editable representation. We expect more methods based on this idea in the near future.

Further, for the purpose of the completeness of this survey, we also briefly mention the approach of **generative modeling** in Section 2.7.

In Section 2.8, we focus on **photogrammetric reconstruction**, which covers another significant body of research work from the computer vision community. However, for compactness and since the focus of our work is image-based methods, we omit pure LIDAR driven approaches, which nonetheless are of significant importance to the field.

Finally, in Section 2.9, we introduce the concepts of **dense reconstruction** based on plane sweep and depth map fusion algorithms.

## 2.2. Façade Image Processing

The problem of processing of urban imagery for reconstruction purposes has been subject of very active research in the recent two decades. Many different approaches for extraction of façade texture, structure, façade elements and façade geometry have been proposed.

**Figure 2.2.:** *A multi-viewpoint panorama of a street in Antwerp composed from 107 photographs taken about one meter apart with a hand-held camera. Figure courtesy of [AAC*06].*

Here we discuss façade texture generation as well as methods for façade segmentation and window detection. We will review methods that cast the façade reconstruction as an traditional image processing and pattern recognition problem or define it as a feature detection challenge.

Recently there are novel methods that cast the problem as a global one and usually propose grammars in order to fit a top-down model. They provide (stochastic) optimization solvers in order to derive the parameters of the model from the façade data. This kind of segmentation algorithms is referred to as *inverse procedural modeling* and we review them in detail in Section 2.6.

### 2.2.1. Panorama Imaging

Panoramas are traditionally generated for the purpose to picturise wide landscapes or similar sights. In praxis, panoramas are composed out of several shots taken at approximately the same location [Sze06].

For urban environments, often the composed image is generated along a path of camera movement, referred to as strip panorama. The goal of those methods is to generate views with more than one viewpoint in order to provide novel insights into the given data. One such variant are pushbroom images, which are orthogonal along the horizontal axis [GH97, SK03], and the similar x-slit images presented by Zomet *et al.* [ZFPW03]. Similar approaches for generation of strip-panoramic images have been proposed also by Zheng [Zhe03] and Roman *et al.* [RGL04]. Agarwala *et al.* [AAC*06] aims at the creation of long multi-view strip panoramas of street scenes, where each building is projected approximately orthogonal on the proxy plane. Optimal source images for particular pixels are chosen using a constrained MRF-optimization process. Similarly to the approach of Roman *et al.* [RGL04], it is inspired by the artistic work of Michael Koller [Kol06]. While our approach presented in Chapter 3 shares several ideas with these papers, our focus lies on estimating an orthographic projection, and on the removal of all disturbing occluders, in order to provide high-quality façade texture.

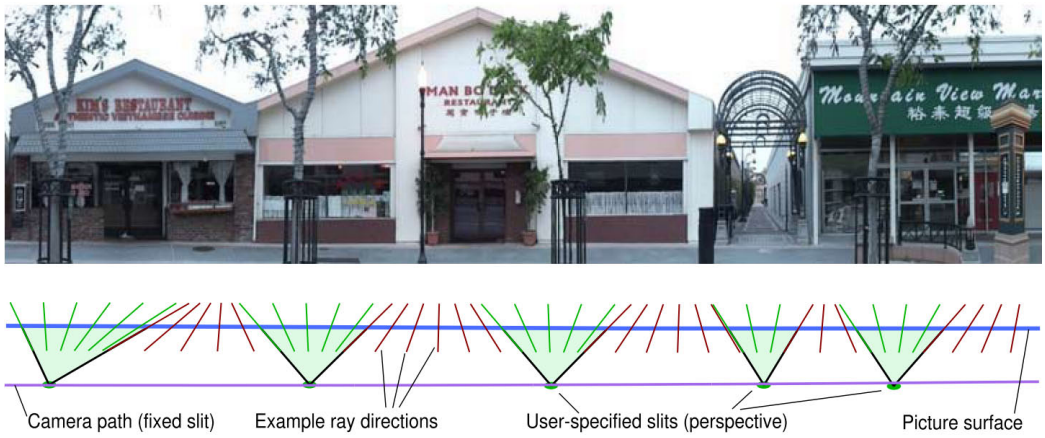


**Figure 2.3.:** *This example shows how manipulation of the perspective structure of the image can be used to generate a multi-perspective image with reduced distortion. The diagram below each is a plan view of the scene, with the input video camera moving along the indicated path and looking upwards. Figure courtesy of [RGL04].*

### 2.2.2. Image Stitching

Stitching of image content from several sources is an old matter, often also referred to as image- or photomosaics. We mention this topic since it has many applications in the field of urban imagery. Especially panoramic images presented in the previous section, as well as projective textures which are described in Section 2.4.2 rely on these techniques. In Chapter 3 we introduce our interactive method which also uses image stitching.

The stitching of two signals of different intensity usually causes a visible junction between them. An early solution to this problem were transition zones and multi-resolution blending [BA83]. Pérez *et al.* [PGB03] introduced a powerful method for this purpose: image editing in the gradient domain. There is a number of further papers tackling, improving, accelerating and making use of this idea [PGB03, ADA*04, Aga07, MP08]. Zomet *et al.* presented an image stitching method for long images [ZLPW06]. Recently, McCann *et al.* [MP08] introduced an interactive painting system which allows the user to paint directly in the gradient domain, and the Poisson equation is solved online by a GPGPU solver. Also Jeschke *et al.* proposed a real-time solver [JCW09]. The foundations behind the gradient domain image editing method are described in the aforementioned papers as well as in the ICCV 2007 Course-Notes [AR07]. For the completeness, we shall provide a brief overview of this approach in Section 3.3.4.

### 2.2.3. Symmetry Detection

Symmetry is abound in typical architecture which is mostly the result of economical, manufacturing as well as aesthetical reasons. Some recent approaches exploit this and try to

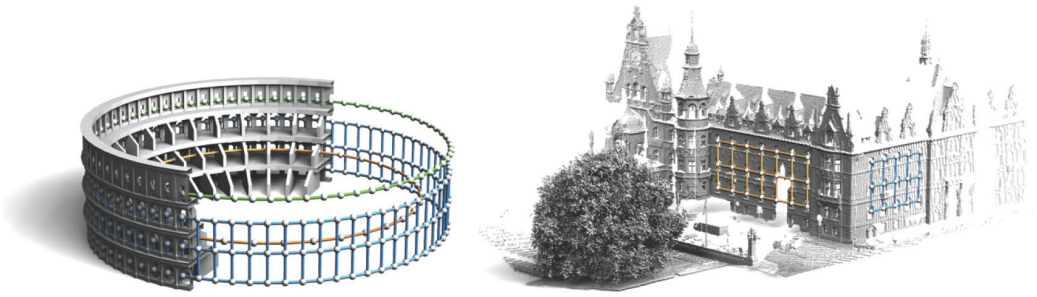

**Figure 2.4.:** *This example shows automatic symmetry detection results performed on point-clouds of architectural objects. Figure courtesy of [PMW*08].*

detect the inherent symmetry in order to infer some information about structure of the façade. Symmetry in general is a topic that inspires researchers since the year one.

In image processing, early attempts have been done by [RWY95] who introduced continuous symmetry transform for images. Next, a considerable load of work on this topic has been done by Liu and collaborators [LCT04]. They detect crystallographic groups in repetitive image patterns using a dominant peak extraction method from the autocorrelation surface. Further succeeding approaches specialize on detecting affine symmetry groups in 2d [LHXS05, LE06] or 3d [MGP06, PSG*06]. Recent follow-ups of these approaches introduce data-driven modeling methods like symmetrization [MGP07] and 3d lattice fitting [PMW*08]. Further, recent image processing approaches tend to utilize the detected symmetry of regular [HLEL06] and near-regular patterns [LLH04, LBHL08] in order to model new images.

### 2.2.4. Repetitive Patterns

Another important research direction, which is often referred to in image-driven urban reconstruction, is the 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. Hsu *et al.* [HLL01] use wavelet decomposition of the autocorrelation surface to segment a regular texture image into tiles.

Turina *et al.* [TTvG01, TTMvG01] 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 façades but there is no application for urban reconstruction using this approach yet.

Boiman and Irani [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.

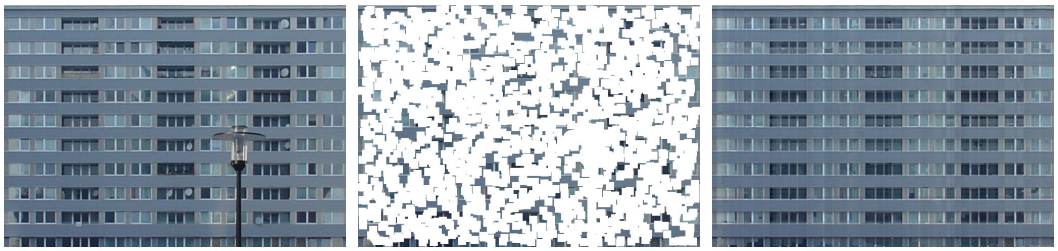

**Figure 2.5.:** *Façade in-painting. The left image is the original image. Middle: the lamp and satellite dishes together with a large set of randomly positioned squares has been selected as missing parts (80% of the façade shown in white). The right image is the result of the tensor completion algorithm proposed in [LMWY09].*

### 2.2.5. Image Factorization

Matrix factorization is a topic of lineaer algebra for a long time [Str05], but just recently is has become interesting in the context of façade imagery. This is due to the fact that images are usually stored in matrix form in computer memory. For this reason it is an obvious idea to apply matrix factorization algorithms on them.

One, well known matrix factorization algorithm is the *singular value decomposition* (SVD) [Str05]. It allows to express a matrix as a weighted product of basis functions that are a set of orthonormal vectors sorted by the variance of the data. In other words, the coordinate system around the data is chosen in such a way, that the information can be expressed by a minimal (optimal) number of bases in the least squares sense. This kind of representation of image data allows e.g., for efficient compression of images of low-rank [HTF09].

Façade images are known to be low-rank, which means that they can be approximated quite well with only view basis functions of the SVD. The approach presented by Ali *et al.* [AYRW09] makes an advantage of this fact and exploits it in order to render massive urban models. They introduce a compression algorithm in order to overcome a memory transfer bottleneck and to render the models from a compressed representation directly. For this purpose they also provide a binary factorization algorithm.

Liu, Musialski and colleagues [LMWY09] propose an algorithm to estimate missing values in tensors of visual data, where a tensor is a generalization of the matrix concept. Their algorithm is built on studies about matrix completion using the matrix trace norm and relaxation techniques to achieve a globally optimal solution. Façade data is well suited for such algorithms due to many repetitions (see Figure 2.5).

### 2.2.6. Segmentation

Most earlier works, which aim in façade structure detection, are based on morphological segmentation and locally acting filtering methods.

The first paper to list here is the work of Wang *et al.* [WTT*02] who concentrate on the appearance of the textured models and the detection of windows. They introduce a façade texture that is based on the (weighted) average of several source images projected on the block-model (called *consensus texture façade* or CTF). This texture serves for both texturing and a source for detection of further detail, like windows (called in the paper "microstructure"). They propose an oriented-region-growing approach (ORG) that is based on iterative enlarging of small seed-boxes until they best fit windows in the CTF. In order to synchronize the boxes they introduce a periodic pattern fixing algorithm (PPF). The assumption that windows are darker than their surrounding façade is, however, weak and may work well only for airborne pictures. Ground based photography 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.* [TLLH05] who calculate a greenness index (GI) to identify and suppress occlusions by vegetation on their façade textures which they extract from drive-by video recordings. They replicate the features along the detected mirroring axes in order to remove occlusions by vegetation. On the cleaned textures they also apply ORG to find dark window regions. They proposed also further extensions to their method to process video input [TLH06, TCLH06]. They use corner detection in order to find interest points over video frames and register them to each other. Then, a common, rectified texture is generated from the input.

Lee and Nevatia [LN04] propose a window detection method that uses only edges. They project the edges horizontally and vertically to get the marginal edge pixel distributions from which they infer the façade subdivision. 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. Also in this thesis we present a novel symmetry-based segmentation method for semi-regular façades in Chapter 4.

Unsupervised segmentation is also the topic of Burochin *et al.* [BTP09]. In this paper the authors present a method for hierarchical, recursive splitting of façade elements in horizontal and vertical direction. This is applied on a "calibrated" façade image, which means that the camera parameters of the actual photograph are known. The image is subdivided according to the radiometric properties by minimizing edge energies. Such kind of global splitting appears to be a promising approach for pre-processing tasks of urban images.

### 2.2.7. Window Detection

Many methods rely on template matching to model windows and other façade detail. The advantage of template matching is that reconstruction results look very realistic. On the other hand, the disadvantage is that results are in most cases not authentic because there is no template database that contains all possible shapes.

Schindler and Bauer [SB03] match shape templates against point clouds. Also Mayer and Reznik [MR07] efficiently match template images from a manually constructed window

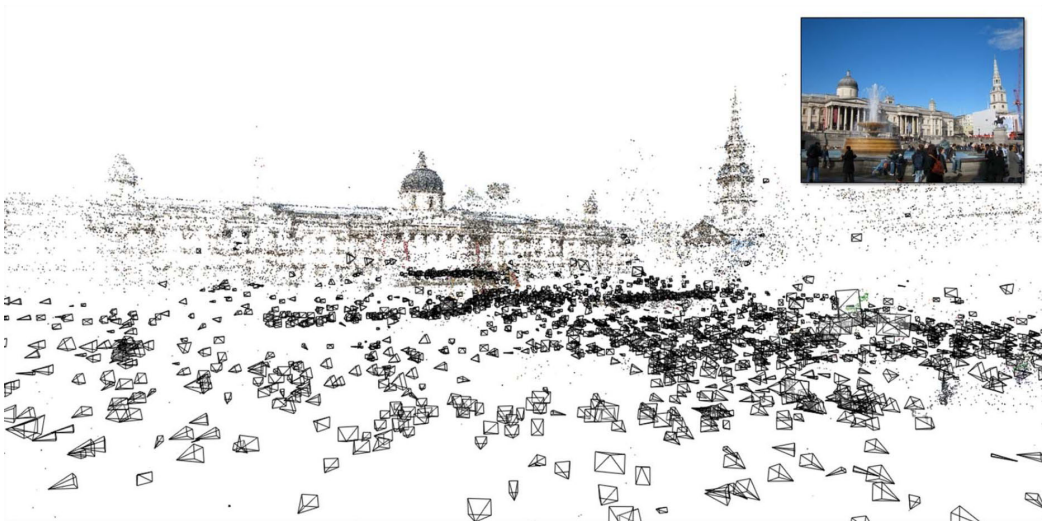

**Figure 2.6.:** *Comparison of a photo to a sparse reconstruction of a huge number of unordered photographs. Figure courtesy of [SSG*10].*

image database against their façades. Müller *et al.* [MZWvG07] match appearance of their geometric 3d window models against façade tiles. Haugeard *et al.* [HPFP09] introduce an algorithm for inexact graph matching, which is able to extract a window as a sub-graph of the graph of all contours of the façade image. This serves as an basis to retrieve similar windows from a database of images of façades.

Some have also combined template matching with machine learning, like Ali *et al.* [ASJ*07] who propose to train a classifier or [DF08] who uses Adaboost [SS99], such that it identifies a high percentage of windows even in images with perspective distortion. Their attempt allows to find appropriate features in order to detect windows automatically in rectified images.

Another approach, which is based on rectangle detection, is the window-pane detection algorithm by Cech and Sara [CS08] that identifies strictly axis-aligned rectangular pixel configurations in a MRF. Given the fact that the majority of windows and other façade elements are rectangular, a common approach to façade 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.

## 2.3. Sparse Reconstruction

In recent years, there has been a considerable trend to feature-based *sparse multi-view stereo* algorithms, which aim at fully automatic sparse reconstruction. The advantage of

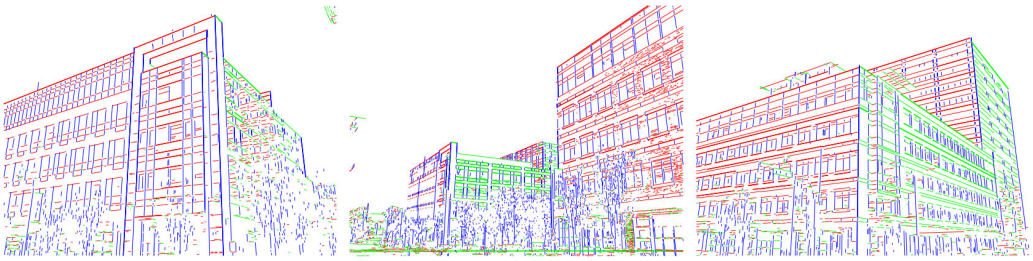

**Figure 2.7.:** *Line-Based Structure from Motion: 2d line features are automatically detected by grouping edge pixels according to vanishing directions. Figure courtesy of [SKD06].*

this approach is its ability to deliver quite accurate extrinsic camera parameters (orientation of the cameras to each other) and also, especially the newer methods, approximated intrinsic parameters (the focal length, the principal point and also the skew factor) of the cameras. Often sparse systems use a *structure-from-motion* methodology, since video sequences are used as input as well. However, sparse systems have been extended to pure image-based methods, where the time-coherence does not play any role. The output of sparse methods is usually a set of images which are registered with respect to each other, and a sparse, colored point-cloud.

### 2.3.1. Structure from Motion

One of the key inventions in this area are robust feature-point detection algorithms, like SIFT [Low04] or SURF [BETvG08], which allow for efficient pairing of corresponding points across multiple images. As input, multiple photographs are provided to the system, and from each one a sparse set of feature-points is extracted and matched. Once multiple images with corresponding features have been established, the extrinsic (i.e., pose in 3d space) and intrinsic (i.e. focal length) parameters as well as the fundamental matrix of their cameras can be determined [Nis04]. Given camera parameters, fundamental matrix and corresponding feature points, 3d space points can be triangulated* [HZ04]. Robust estimation algorithms (e.g. RANSAC [FB81]) and advanced non-linear bundle-adjustment solvers [TMHF99, LA04] are typically used to compute highly accurate point-clouds of 3d structure out of ordinary photographs. The advantage of this approach is its conceptual

*Note that the term "triangulation" has at least three different meanings depending on the context. In computer graphics "triangulation" usually means that a set of space points (2d or 3d) is topologically connected to a triangle mesh. In contrast, in computer vision the expression "stereo structure triangulation" means that 3d space points are determined from known camera matrices, the fundamental matrix and corresponding point-pairs [HZ04]. This is more related to the meaning used in photogrammetry and remote sensing, as well as in trigonometry and geometry, where triangulation denotes a method of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly. The point can then be fixed as the third point of a triangle with one known side and two known angles.

**Figure 2.8.:** *Results of interactive modeling method presented by Sinha et al. [SSS*08].*

simplicity and robustness. It is quite general, since it is a bottom-up approach in its nature, and it does not expect any model imposed on the data.

### 2.3.2. Image-Based Sparse Systems

The structure-from-motion method is used to register multiple images to one another and to orient and position them in 3d space. It is based on feature matching, pose estimation, and bundle adjustment. An example is the ARC3D web service [arc10], to which people can upload images and get (sparse or dense) 3d data as well as camera parameters back [VvG06]. Images of buildings are among the most often uploaded data. More recently, similar systems, but more dedicated to city exploration and reconstruction, have been proposed [SSS06,SSS07,SGSS08,GSC*07,IZB07,ASS*09]. We also utilize a similar solution in order to provide registered multi-view input to our method in Chapter 3, Section 3.3.1.

## 2.4. Interactive Modeling

Manual modeling of architecture is a tedious and time consuming task. However, for a long time (in the 80's and 90's) it was the only way to obtain 3d models of urban sites. Debevec *et al.* [DTM96] introduced a hybrid semi-automatic method that combines geometry-based modeling with image-based modeling into one pipeline. This system makes it possible to model 3d geometry under the preservation of epipolar constraints that are computed from a set of perspective photographs of the target scene. In [DTM96] the images are registered to each other by manually establishing corresponding features between them. Since the images are shot from different positions, correspondences allow to establish epipolar-geometric relations between them which can be set up as a non-linear optimization problem.

The geometry of the scane is modeled with polyhedral blocks, which is based on a number of assumptions: (1) Most architectural scenes are well modeled by an arrangement of geometric primitives. (2) Blocks implicitly contain common architectural elements such as parallel lines and right angles. (3) Manipulating block primitives is convenient since

**Figure 2.9.:** *Interactive modeling of geometry in video. Left: Replicating the bollard by dragging the mouse. Right: Replicating a row of bollards. Figure courtesy of [vdHDT*07a].*

they are at a suitably high level of abstraction; individual features such as points and lines are less manageable. (4) A surface model of the scene is readily obtained from the blocks, so there is no need to infer surfaces from discrete features. (5) Modeling in terms of blocks and relationships greatly reduces the number of parameters that the reconstruction algorithm needs to recover. These observations turned out to be quite appropriate and have subsequently been applied in several other automatic and semi-automatic approaches.

### 2.4.1. Modeling with Epipolar Constraints

Debevecs seminal paper can be seen as a starting point of a series of follow-up works which deserve to be classified with the term *image-based modeling*. Inspired by this work, there have been a number of methods based mainly on the strict assumptions of the epipolar geometry across a number of perspective images. These provide well formulated geometric problems that are solved by (non-linear) optimization. The most commonly used geometric constraints in 3d-multi-view vision are the paradigms of parallelism and orthogonality prevalently present in indoor and outdoor architectural scenes, often detected via the corresponding vanishing points and their layout. Similar methods, which try to solve the problem without user interaction, like Liebowitz [LZ98] and Werner and Zisserman [WZ02], are handled in Section 2.8.

Liebowitz *et al.* [LCZ99] presented a set of methods for creating 3d graphical models of scenes from a limited numbers of images in situations where no scene coordinate measurements are available. The method employs constraints available from geometric relationships, which are common in architectural scenes, such as parallelism and orthogonality, together with constraints available from the camera. Also Cipolla *et al.* [CR99], and Lee and Nevatia [LN03] proposed systems for recovering 3d models from uncalibrated images of architectural scenes based on the observations of the constraints of epipolar geometry.

A series of papers published by van den Hengel and colleagues describes building blocks of an image- and video-based reconstruction system. The method in [vdHDT*06] uses camera


**Figure 2.10.:** *Interactive façade modeling results. Figure courtesy of [XFT*08].*

parameters and point-clouds generated by a structure-from-motion process (Section 2.3) as a starting point for developing a higher level model of the scene. The system relies on the user to provide a minimal amount of structure information from which more complex geometry is extrapolated. The regularity typically present in man-made environments is used to minimize the interaction required, but also to improve the accuracy of fit. They extend their higher level model in [vdHDT*07a], such that the scene is represented as a hierarchical set of parameterized shapes. Relations between shapes, such as adjacency and alignment are specified interactively, such that the user is asked to provide only high level scene information and the remaining detail is provided through geometric analysis of the image set (cf. Figure 2.9). In a follow-up work [vdHDT*07b] they present the VideoTrace-system for interactively generating 3d models that also relies on sketches drawn by the user under the constraints of 3d information obtained from epipolar geometry.

Sinha *et al.* [SSS*08] presented an interactive system for generating textured 3d models of architectural structures from unordered sets of photographs. It is also based on structure-from-motion as the initial information for cameras and structure. This work introduced novel, simplified interactions such as drawing of outlines overlaid on 2d photographs. The 3d structure is then automatically computed by combining the 2d interaction with the multi-view geometric information from structure-from-motion analysis. The system utilizes vanishing point constraints during the reconstruction, which is useful for architectural scenes. The approach enables to accurately model polygonal faces from 2d interactions in one of the input images (cf. Figure 2.8).

Another interactive image-based approach to façade modeling has been introduced by Xiao *et al.* [XFT*08]. It uses images captured along streets and also relies on structure-from-motion (cf Section 2.3) as source for camera parameters and initial 3d data. It considers façades as flat rectangular planes or simple developable surfaces with an associated texture. Textures are composed from the input images. In their system the façades are interactively subdivided in a top-down manner and structured as a graph of rectilinear elementary patches. This is then followed by a bottom-up merging with the detection of reflectional symmetry and repetitive patterns. All tasks are combined with user interaction who is re-

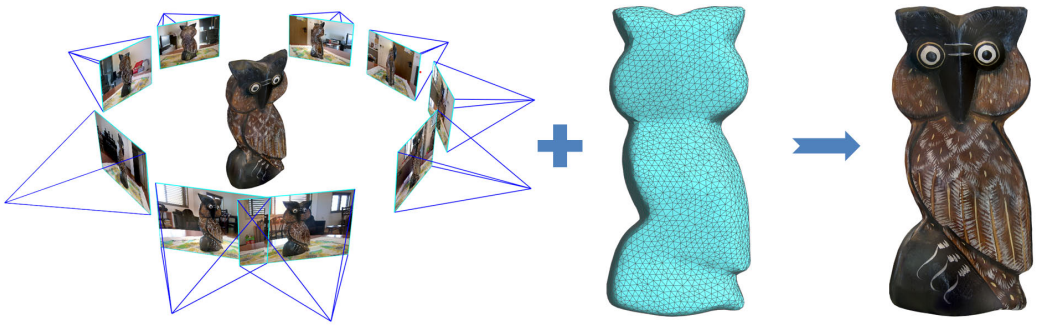

**Figure 2.11.:** *Projective texturing approach. Images registered to each other (including camera parameters) and a 3d model are brought into a common coordinate frame. For each texel the optimal color is specified by optimization algorithms. Figure courtesy of [GWOH10].*

sponsible to permanently correct misinterpretations of the automatic routines (cf. Figure 2.10).

A method to recover 3d models from a single image was presented by Jiang *et al.* [JTC09]. Their approach is similar to those mentioned above but is relies on only one input image. The limitation of this approach is that it only performs for highly symmetric objects, because the epipolar constraints are derived from shape symmetries. Thus they present also a novel algorithm to calibrate the camera from a single image, and introduce a method which allows for 3d points recovery similar to structure-from-motion. These serve again as input for further, interactive modeling and texturing steps which finally result in complete 3d polygonal models. While in general this method is quite interesting, it is very limited in the number of possible objects which can be modeled.

### 2.4.2. Projective Texturing

In the literature the expression *façade texture generation* means mostly the synthesis of a new façade image from one or more photographs. Usually, the goal is to generate an undistorted, rectified and approximately orthogonal image of the façade.

This problem can be addressed by projective texturing of urban sites from perspective photographs. One of the pioneering works was the "Façade" system introduced by Paul Debevec *et al.* [DTM96]. Their paper proposes an interactive modeling tool that allows the user to model 3d architecture from photographs under the constraints of epipolar geometry, and to sample projective textures on building façades. There have been a number of parallel and follow-up publications aiming at urban modeling from images [LZ98, SHS98, CT99, SA02], which utilized the projection of photographs in order to obtain approximated ortho-images. We mention some of theme also in the interactive image-based modeling context in Section 2.4.1.

More recent approaches introduce semi-automatic systems, which support the user during the modeling process. They are based on input from video [vdHDT*07c] or image collections [ARB07,SSS*08,XFT*08] and they introduce texture sampling as part of their modeling pipeline. All these approaches rely on user interaction in order to improve the quality of the results. There are also fully automatic attempts (some of them in the photogrammetry literature) which aim at texture generation for existing models [TKO08, KZZL10]. Recently Xiao *et al.* presented an automated attempt at the modeling and texturing [XFZ*09] of street sites. Kopf *et al.* present a method for photo enhancement by information obtained from a projection on a 3d model [KNC*08].

Tools for interactive, projective texture generation, enhancement, and synthesis for architectural imagery have recently been presented by Pavic *et al.* [PSK06], Korah and Rasmussen [KR07b], Eisenacher *et al.* [ELS08], and Musialski *et al.* [MWR*09, MLS*10]. Another branch are feature-based sparse reconstruction methods, which also make use of projective imaging [SSS06, SSS07, SGSS08].

Finally, there are methods, which do not focus on architecture, but on the problem of projective texturing in general [NK01, PDG05, LI07, TS08, TBTS08, GWOH10]. All of these methods are based on a per-pixel sampling of appropriate color values from a set of registered images (cf. Figure 2.11).

Image-based rendering is related to our work as well: Debevec *et al.* introduced such a system in [DYB98] that was followed by other involved image-based rendering methods [BBM*01, EDM*08]. These approaches aim more at real-time performance than at high-quality images.

## 2.5. Procedural Modeling

Procedural architectural modeling is an approach to model urban environments by the means of rules defined by production systems, like Chomsky grammars [Sip96], L-systems [PL90, PHT93], shape grammars [Sti75] or set grammars [WWSR03]. In architecture, Stiny pioneered the idea of shape grammars [Sti75]. These shape grammars were successfully used for the construction and analysis of architectural design. The reader shall refer to the recent comprehensive state-of-the-art report of the Eurographics by Vanegas *et al.* [VAW*10] for a comprehensive review of this matter.

In this section we provide a brief overview over the literature which aims at synthesis of urban models, but we limit ourself only the seminal works in the area.

### 2.5.1. City Generation Systems

The generation of whole city models is a matter which has become active with the emerge of virtual worlds like computer games. Other application synthesized cities are e.g. simula-

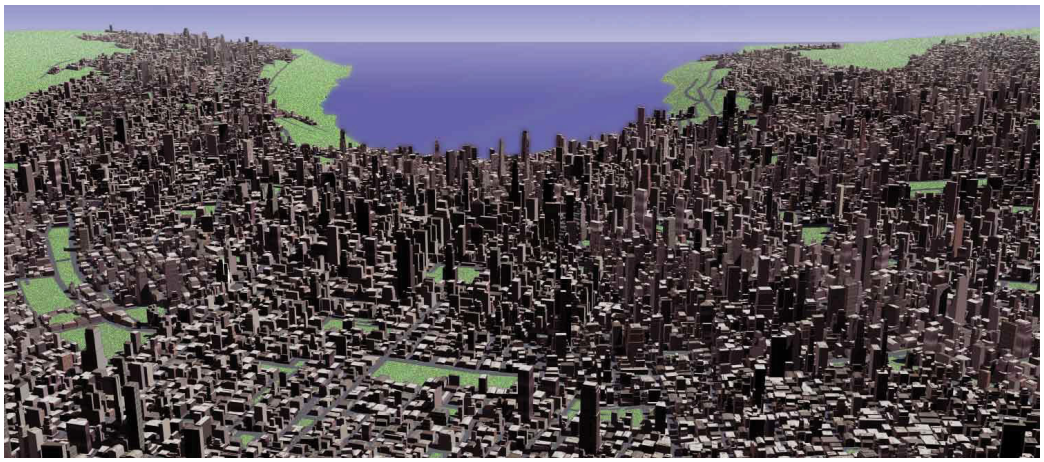

**Figure 2.12.:** *A virtual city modeled automatically. Approximately 26000 buildings were created. Figure courtesy of [PM01].*

tion and urban planning. In general, the goal is here not to reconstruct given environments, but to generate plausible city models, usually including texture and fine level details.

One of the first papers that proposes automatic generation of whole urban environments is the work of Parish and Müller [PM01]. Their system uses a procedural approach based on L-systems to model cities. From various input image maps, such as land-water boundaries and population density, they generate highways and streets, land lots, and compute the geometry for the model. This work can be seen as pioneering in this scope and it was the starting point of the software know as the CityEngine [Mül10] system.

Further work on this topic focus on refinement and extension of the original idea. Interactive modeling of whole cities with improved algorithms for street network generation as well as flexible user modeling operations is proposed by Chen *et al.* [CKX*08]. Interactive reconfiguration of cities as well as connection to real Geographic Information Systems (GIS) is handled in Aliaga *et al.* [ABVA08].

Vanegas *et al.* [VABW09] aim in more compete urban design which includes, except the geometric, also the behavioral modeling of urban spaces. Their system provides an interactive city modeling interface that interacts with an iterative optimization process in order to reaching equilibrium state during modeling. This allows to create realistically looking data for huge cities which are then generated by the above mentioned CityEngine within seconds.

### 2.5.2. Procedural Modeling of Buildings

In order to make the shape grammars better suitable for modeling, Parish, Müller, Wonka and others [PM01, WWSR03, MVW*06, MWH*06, WMV*08] introduced a procedural ar-

**Figure 2.13.:** *Variations of a building modeled procedurally by the method presented by Müller et al. [MWH*06].*

chitecture generation framework. In such systems, buildings are modeled using two interlocked grammars and an attribute matching system. The shape grammar (also called split grammar) starts from an initial shape and splits this shape recursively. The splits to be applied and the symbols to be chosen are determined in a separate ”design“ grammar, the control grammar, which is invoked for each rule selection step. Furthermore, the control grammar refines the attributes in the grammar during the derivation . Figure 2.14 shows a result generated by he system proposed by Wonka *et al.* [WWSR03]. The system is quite complex, and it is difficult to design the rules for both the split grammar and the design grammar so that they work together seamlessly. This problem has been partially resolved by interactive rule editing tools like presented by Lipp *et al.* [LWW08].

In another work, Marvie *et al.* [MPB05] presented a FL-system, which is an extension of a L-system, that allows to generate any kind of object hierarchy on the fly. It is a modification of the classical L-system rewriting mechanism that produces a string of symbols interpreted afterwards. The authors show that thanks to this extension, their approach is able to simulate all of the existing solutions proposed by classical L-systems, as well as to generate VRML97 scene graphs and geometry.

Whiting *et al.* [WOD09] proposed a system that is able to procedurally model a specific architectural style: masonry buildings. Their approach takes additional constraints into account, like the knowledge of masonry style. Moreover, it introduces structural feasibility into procedural modeling. This allows for more realistic structural models that can be interacted with in physical simulations.

Also Merrel and Manocha [Mer07, MM08, MM09] provide a system for generation of architecture based on procedural modeling and additional constraints. Further, it is an example based system, such that user-provided examples are analyzed in the first stage and in the next stage rules and geometry are synthesized accordingly.

Finally, we shall mention the work presented by Finkenzeller [Fin08]. He introduced a method for manual modeling and modifying of detailed building façades that adapt to geometry automatically, such that the user can modify the façade’s outline, window placement, and different styles on an abstract level. He makes use of predefined tree-representations of buildings structure which allows the user to model by specifying the architectural style and thus the appearance. The system generates then the geometry. The

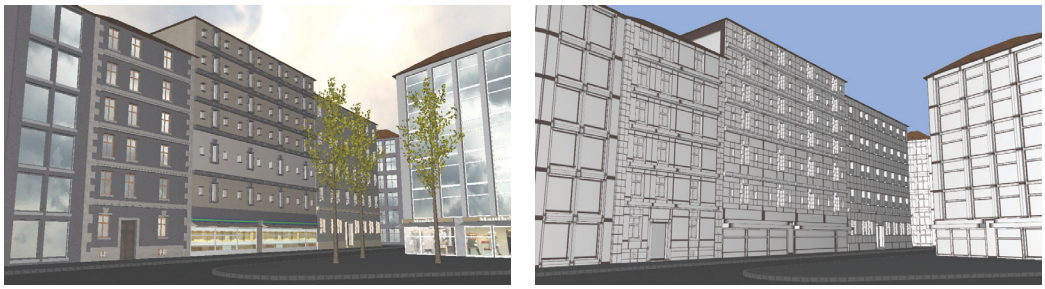

**Figure 2.14.:** *Left: This image shows several buildings generated with split grammars, a modeling tool introduced in this paper. Right: The terminal shapes of the grammar are rendered as little boxes. A scene of this complexity can be automatically generated within a few seconds. Figure courtesy of [WWSR03].*

limitation of this methods is the complicated way of the creation of the rules and architectural styles.

## 2.6. Inverse Procedural Modeling

In the previous chapter we have introduced the concept of procedural modeling. It provides an elegant and fast way to generate huge, complex and realistically looking urban sites. Due to its generative nature it can also be referred to as *forward procedural modeling*. A recent survey [VAW*10] presents this approach for synthesis of urban environments.

### 2.6.1. Inverse Modeling of Buildings

On the other hand, the paradigm of grammar driven building model construction is not limited only to pure synthesis, but also to the reconstruction of existing buildings. A very complete, yet manual solution to this problem has been presented by Aliaga *et al.* [ARB07]. This paper presents an *inverse procedural modeling* system for whole urban buildings. They extract manually a repertoire of grammars from a set of photographs of a building and utilize this information in order to visualize a realistic and textured urban model. This approach allows for quick modifications of the architectural structures, like number of floors or windows in a floor. The disadvantage of this approach is the quite labor intensive grammar creation process.

Another approach to inverse procedural modeling has been recently proposed by Bokeloh *et al.* [BWS10]. This work aims in slightly a different goal: automatic extraction of grammars by the means of an exemplar geometric model. Further the paper discusses the idea of a general rewriting system and context free rules for geometry, thus it provides important cues to the still very novel research topic.

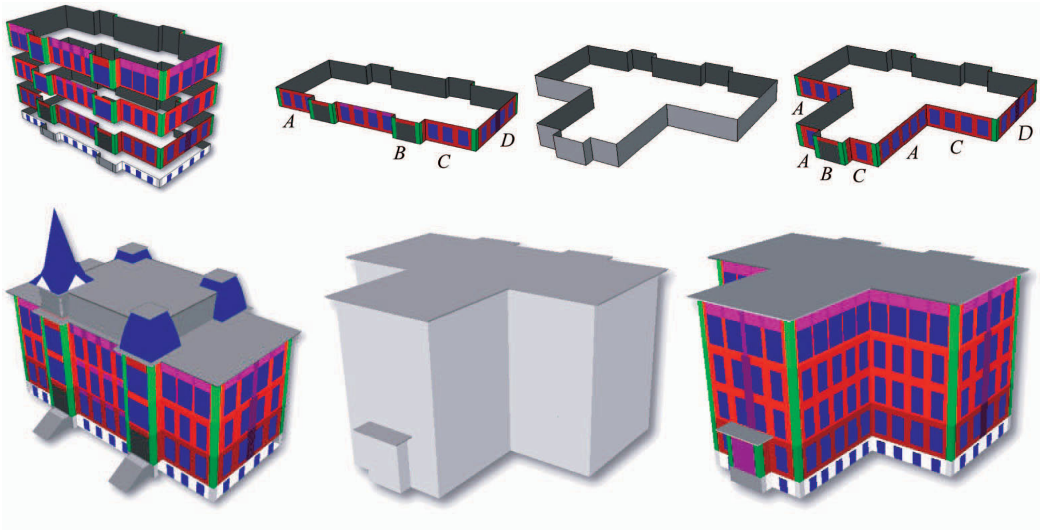


**Figure 2.15.:** *Example of inverse procedural modeling. Figure courtesy of [ARB07].*

Also Vanegas *et al.* [VAB10] proposed a method to extract block models of buildings from images based on a grammar and Stava et al. a technique to infer a compact grammar from arbitrary 2d vector content [SBM*10].

The paper of Dick *et al.* [DTC04] describes an automatic acquisition attempt of three dimensional architectural models from short image sequences. The approach is Bayesian and model based. Bayesian methods necessitate the formulation of a prior distribution; however designing a generative model for buildings is a difficult task. In order to overcome this a building is described as a set of walls together with a "Lego" kit of parameterized primitives, such as doors or windows. A prior on wall layout, and a prior on the parameters of each primitive can then be defined. Part of this prior is learnt from training data and part comes from expert architects. The validity of the prior is tested by generating example buildings using *Markov Chain Monte Carlo* (MCMC) and verifying that plausible buildings are generated under varying conditions. The same MCMC machinery can also be used for optimizing the structure recovery, this time generating a range of possible solutions from the posterior. The fact that a range of solutions can be presented allows the user to select the best when the structure recovery is ambiguous.

A general work which aims on grammar driven segmentation has been published by Han and Zhu [HZ05, HZ09]. It presents a simple attribute graph grammar as a generative representation for made-made scenes and proposes a top-down/bottom-up inference algorithm for parsing image-content. Is simplifies the objects which can be detected to quare boxes in order to limit the grammar space. Nevertheless, this approach provides a good starting point for inverse procedural image segmentation.

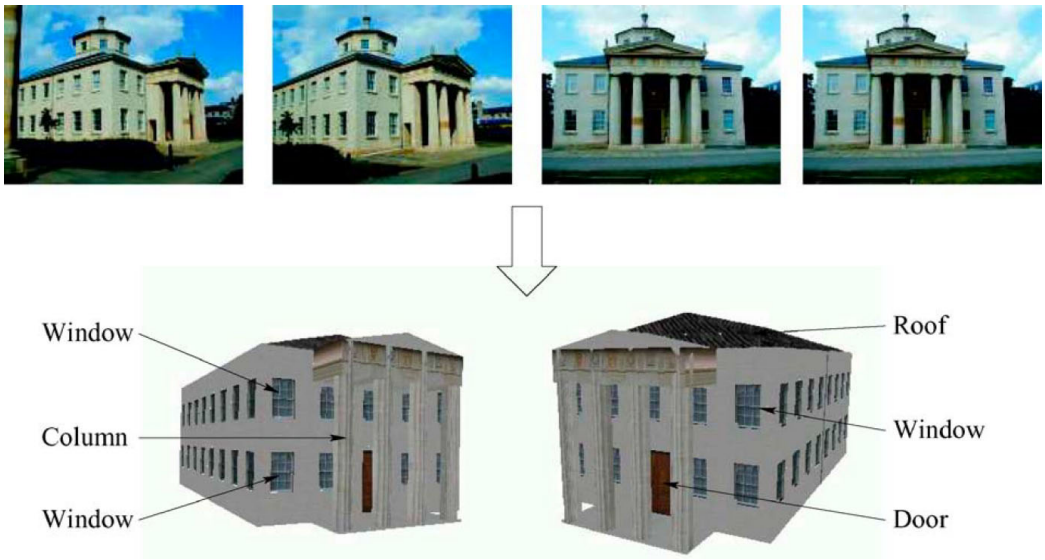


**Figure 2.16.:** *Example of inverse procedural modeling: A labeled 3d model is generated from several images of an architectural scene. Figure courtesy of [DTC04].*

### 2.6.2. Inverse Modeling of Façades

It appears plausible to adapt the concept of inverse procedural modeling to reconstruct façades. In this section we discuss the class of solutions that are driven by hierarchical, rule based segmentation algorithms. They cut down a façade into small irreducible parts which are arranged according to hierarchical context free grammar rules. A single-view approach for rule extraction from segmentation of simple regular façades has been published by Müller *et al.* [MZWvG07] who cut the façade image into floors and tiles. The tiles are then synchronized, split and finally procedural rules are extracted.

However, already Alegre and Dellaert [AD04] proposed a specific set of grammar rules and a Markov Chain Monte Carlo (MCMC) approach to optimize the parameters in order to fit the hierarchical model against the façade image. Yet, the model they provide does not generalize to a large class of building façades. Also Korah and Rasmussen introduced a method for automatic detection of window-grids in ortho-rectified façade images [KR07a] based on MCMC optimization. Further, Mayer and Reznik [MR05, MR06, MR07, RM07, May08] propose a series of papers, where they present a system for façade reconstruction and window detection by fitting a model by MCMC. Van Gool *et al.* [vGZBM07] search for similarity chains in perspective images to identify repeated façade elements. Hohmann *et al.* [HKHF09] attempts an interactive solution combined with a façade grammar.

Brenner and Ripperda [BR06, RB07, Rip08, RB09] develop in a series of publications a system for grammar-based decomposition of façades in elements from images and laser scans. Also they utilize MCMC for optimization. The papers of Becker *et al.*

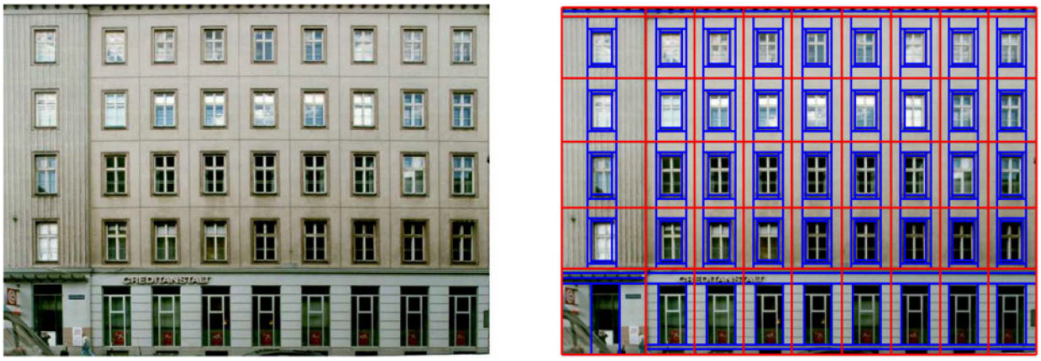

**Figure 2.17.:** *Left: original image and right: segmentation produced by the method of [MZWvG07].*

[BH07, BHF08, Bec09, BH09] and Pu and Vosselman [PV09a, PV09c] are about building and detailed façade reconstruction from photographs and LIDAR scans by utilizing higher order models.

A recent approach [KST*09] examines a rectified façade image in order to fit a hierarchical tree grammar. This task is formulated as a *Markov Random Field* (MRF) [GG84] and solved by an approximating algorithm. In the following, the tree formulation of the façade image is converted in to a shape grammar which is responsible to generate a model in procedural modeling style. Teboul *et al.* [TSKP10] extend their work by combining a bottom-up segmentation through superpixels with top-down consistency checks coming from style rules. The space of possible rules is explored efficiently.

## 2.7. Generative Modeling

In the context of modeling of urban geometry we shall also mention the approach developed by Sven Havemann [Hav05]. He proposes a novel model representation method. Its main feature is that 3d shapes are represented in terms of functions instead of geometric primitives. Given a set of typically only a few specific parameters, evaluating such a function results in a model that is one instance of a general shape. The shape description language (GML) is a full programming language, but it has an quite simple syntax. It can be regarded as some form of a "mesh creation/manipulation language". It is designed to facilitate the composition of more complex modeling operations out of simpler ones. Thus, it allows to create high-level operators which evaluate to arbitrarily complex, parameterized shapes.

**Figure 2.18.:** *A collection of rendered images from the final 3d city model taken from various vantage points. Figure courtesy of et al. [CLCvG07].*

## 2.8. Photogrammetric Reconstruction

This section provides an overview of automatic approaches for reconstruction of urban architecture. The common property of these is the demand on minimal user interaction or even, in the best case, no user-interaction at all. There is quite a variety of approaches, which either work with aerial or ground-level input data. It is difficult to compare these methods directly to each other since they have been developed in different contexts (types of input data, types of reconstructed buildings, level of interactivity, etc.).

Many systems up to the year 2003 have been also reviewed in a comprehensive survey by Hu *et al.* [HYN03]. Due to the imagery-related topic of this report, we limit ourselves to methods that expect image data as (at least partial) input and omit these which work purely with LIDAR data.

### 2.8.1. Ground Based

Pollefeys *et al.* [PvGV*04] presented an automatic system to build visual models from images. This work is also one of the papers which pioneers fully automatic structure-from-motion of urban environments. The system can deal with uncalibrated image sequences acquired with a hand-held camera and is based on features matched across multiple views. From these both the structure of the scene and the motion of the camera are retrieved.

A ground-level city modeling framework which integrates two components, reconstruction and object detection has been presented by Cornelis *et al.* [CLCvG07]. It proposes a highly optimized 3d reconstruction pipeline that can run in real-time, thereby offering the possibility of online processing while the survey vehicle is recording. A realistically textured, compact 3d model of the recorded scene can already be available when the survey vehicle returns to its home base. The second component is an object detection pipeline, which detects static and moving cars and localizes them in the reconstructed world coordinate system.

The paper of Irschara *et al.* [IZB07] provides a combined sparse-dense method for city sites reconstruction from unstructured photo-collections. Their work uses images contributed by end-users as input. Hence, the Wiki principle well known from textual knowl-


**Figure 2.19.:** *Result of the automatic method proposed by Xiao et al. [XFZ*09].*

edge databases is transferred to the goal of incrementally building a virtual representation of the occupied habitat. In order to achieve this objective, state-of-the-art computer vision methods, such as structure-from-motion and dense matching are applied and modified accordingly.

Recently, Xiao *et al.* [XFZ*09] attempt to extend their previous method [XFT*08] in order to provide an automatic approach to generate street-side 3d photo-realistic models from images captured along the streets at ground level. They propose a multi-view semantic segmentation method that recognizes and segments each image at pixel level into semantically meaningful areas, each labeled with a specific object class, such as building, sky, ground, vegetation and car. A partitioning scheme is then introduced to separate buildings into independent blocks using the major line structures of the scene. Finally, for each block, they propose an inverse patch-based orthographic composition and structure analysis method for façade modeling that regularizes the noisy and missing reconstructed 3d data. The system has the advantage of producing visually compelling results by imposing strong priors of building regularity. The price the method pays for the automatization is the clearly visible quality loss when compared to [XFT*08] as can be seen in Figures 2.10 and 2.19.

Furukawa and Ponce [FP07,FP09] presented a novel approach for multi-view stereo reconstruction. This method is based on small patches, which are optimized in order to determine 3d structure. This basically generic 3d reconstruction method has been extended and applied to 3d urban reconstruction in [FCSS09a]. and also successfully extended to reconstruct interiors [FCSS09b]. Recently, they introduced also large scale city reconstruction approach [FCSS10] based on the same methodology.

There are several other, outstanding contributions which aim at fully automatic reconstruction from ground-based imagery, e.g., Teller [Tel98], Stamos and Allen [SA00,SA01, SA02], Rother and Carlsson [RC02], Schindler and Bauer [SB03], Bauer *et al.* [BKS*03], Kosecka and Zhang [KZ05], and recently also the method of Akbarzadeh *et al.* [AFM*06] as well as Pollefeys *et al.* [PNF*07],

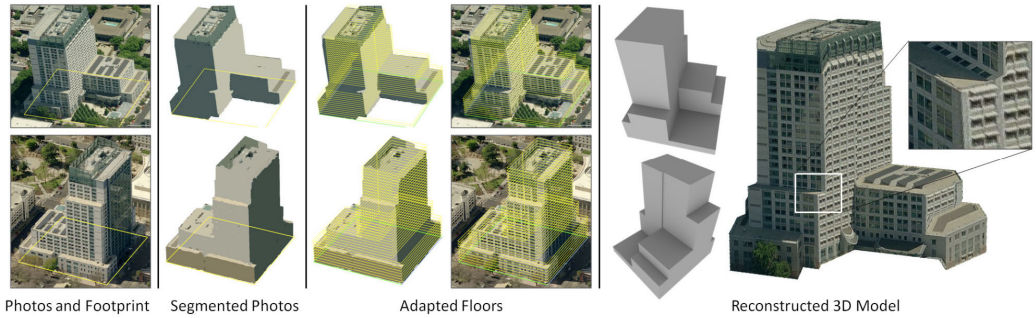


**Figure 2.20.:** *Results of the hybrid method which uses aerial imagery registered to maps and an inverse procedural grammar. Figure courtesy of Vanegas et al. [VAB10].*

### 2.8.2. Aerial and Hybrid

Besides reconstruction of terrestrial imagery as presented in the previous section, there is a considerable body of work done on reconstruction of aerial images and LIDAR scans. There is also a number of approaches that combine terrestrial and aerial images such as the work of Wang *et al.* [WYN07].

Further, there are approaches to combine imagery with LIDAR, such as the work of Früh and Zakhor, who published a series of articles that aim at a fully automatic solution for large scale urban reconstruction. First they propose an approach for automated generation of textured 3d city models with both high details at ground level, and complete coverage for bird's-eye view [FZ03]. A close-range façade model is acquired at the ground level by driving a vehicle equipped with laser scanners and a digital camera under normal traffic conditions on public roads. A far-range digital surface model (DSM), containing complementary roof and terrain shape, is created from airborne laser scans, then triangulated, and finally texture-mapped with aerial imagery. The façade models are first registered with respect to the DSM using Monte Carlo localization, and then merged with the DSM by removing redundant parts and filling gaps. In further work [FZ04] they improve their method for ground-based acquisition of large-scale 3d city models. Finally, they provide a comprehensive work which introduces a set of data processing algorithms for generating textured façade meshes of cities from a series of vertical 2d surface scans and camera images [FJZ05].

Also the work done by Pu and Vosselman [PV09a, PV09b, PV09c] is mainly about building and façade reconstruction from point-clouds. Laser data and optical data have a complementary nature for three dimensional feature extraction. Efficient integration of the two data sources will lead to a more reliable and automated extraction of three dimensional features.

Mastin *et al.* [MKF09] proposed a method for fusion of 3d laser radar (LIDAR) imagery and aerial optical imagery in order to construct 3d virtual reality models. They utilize the well known downhill simplex optimization to infer camera pose parameters and discuss

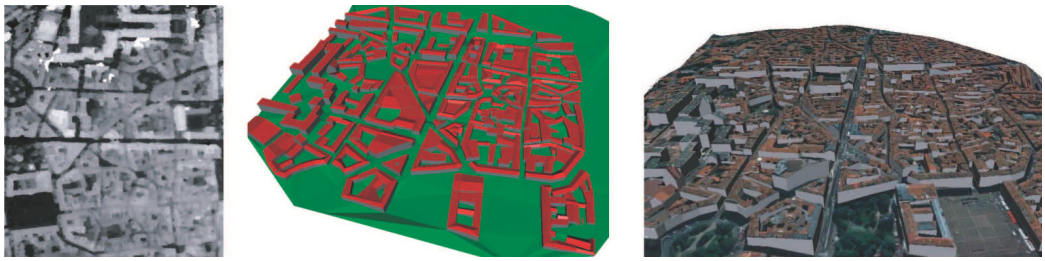

**Figure 2.21.:** *Automatic urban area reconstruction results from a DSMs (left): without (middle) and with textures (right). Figure courtesy of [LDZPD10].*

three methods for measuring mutual information between LIDAR imagery and optical imagery and use OpenGL and graphics hardware in the optimization process, which yields registration times lower than previous methods.

Recently, there have been quite a number of publications in the computer vision literature which involve several types of input data besides conventional ground based photographs. In particular, we refer the reader to methods which work with aerial imagery, like Jaynes *et al.* [JRH03], Zebedin *et al.* [ZBKB08], Poullis and You [PY09], Vanegas *et al.* [VAB10] or Karantzalos and Paragios [KP10], and Lafarge *et al.* [LDZPD10], as well as with maps and geo-references, like Georgiadis *et al.* [GSGA05], El-hakim *et al.* [EhWGG05], Pollefeys *et al.* [PNF*07] and Grzeszczuk *et al.* [GKVH09].

## 2.9. Dense Reconstruction

In computer vision the term *dense matching* is generally used for image-based reconstruction of detailed surfaces as shown in Figure 2.22. In this context, dense denotes that such systems try to capture information from all pixels in the input images – in contrast to sparse methods (cf. Section 2.3) where only selected feature points are considered. One of the most reliable methods for this task is denoted as the he *plane sweep* approach proposed by Collins [Col96]. It has been considerably extended in recent years [BZB06, GFM*07] in order to perform with modern programmable hardware graphics accelerators [YP03]. In general, this approach is based on discretization of the target space into a grid. Than, a plane is swept discreetly through this space along one of its axes and rays are shot from all pixels of all cameras onto the plane. According to epipolar geometry, intersections of the rays with each other at their hitpoints on the plane indicate 3d structure points. The plane sweep method allows to efficiently accumulate these points and to generate dense surface reconstructions [Col96, BKS*03].

Dense structure of the surface is also computed by a multi-view stereo matching algorithm proposed by Pollefeys [PvGV*04]. Vergauwen and Van Gool [VvG06] extended this method from regular sequences of video frames to still images by improved feature match-

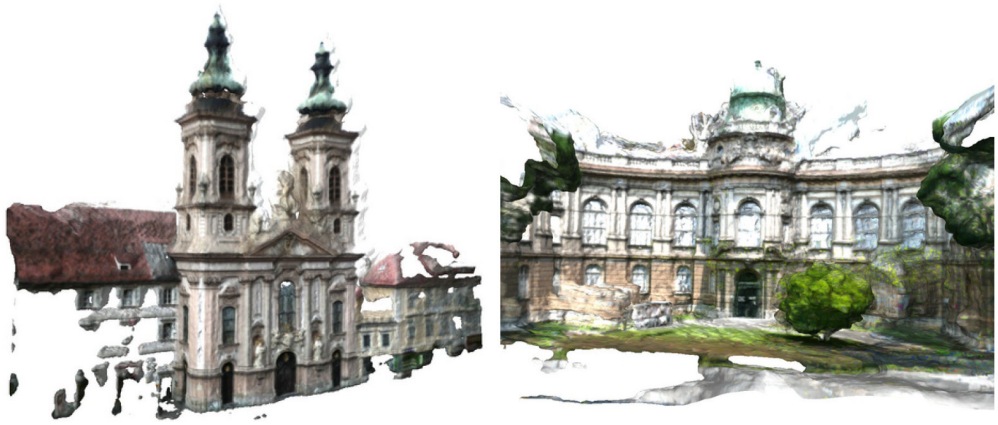

**Figure 2.22.:** *Examples of dense reconstruction after depth map fusion. Figure courtesy of [IZB07].*

ing, additional internal quality checks and - at a time where EXIF data were less widespread - methods to estimate internal camera parameters. This approach was introduced as the free, public ARC3D web-service, allowing the public to take or collect images, upload them, and get the result as dense 3d data (and camera calibration parameters).

Another approach for the reconstruction of dense structures is to perform pairwise dense matching of any two to each other registererd views and than to combine the computed depth-maps with each other. Usually this approach is denoted as *depth map fusion*. There are several ideas how to perform this, such as Goesele *et al.* [GCS06], Zach *et al.* [ZPB07, IZB07], Merrell *et al.* [MAW*07] and finally the recent novel approach by Micucik and Kosecka [MK10].

Generally, recent approaches deliver quite impressive results (see Figure 2.22) on the one hand, on the other, these systems usually provide dense polygonal meshes without any higher-level knowledge of the underling scene. This stays in contrast to the inverse procedural methods mentioned in Section 2.6. We are hopefully to expect approaches which try to grasp the best of both worlds in order to provide dense structure combined with semantic information in future.

## 2.10. Summary

Urban modeling and reconstruction is a very wide topic. This review tries to provide a comprehensive overview of the most important concepts in this field from the point of view of computer graphics research.

First of all we want to argue that the problem of image stitching and blending has reached a very mature stage – especially with the introduction of gradient domain methods. This

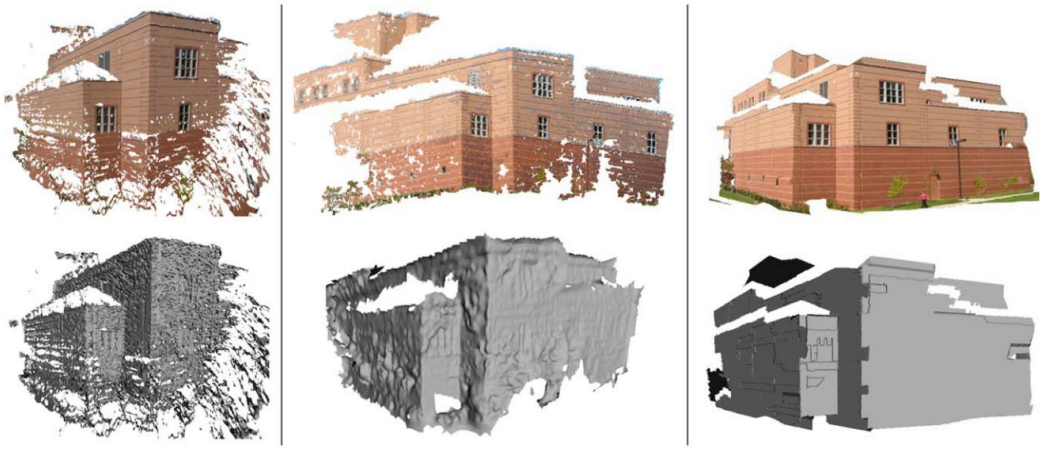

**Figure 2.23.:** *Comparison of 3d models created by different methods. Left: Vergauwen and van Gool [VvG06], middle: Furukawa and Ponce [FP07], right: Micusik and Kosecka [MK10]. Figure courtesy of [MK10].*

research topic of image processing helps a lot in the field of urban reconstruction, which depends on high qualitative input imagery.

Regarding multi-view image processing, image registration, and especially structure-from-motion algorithms we can also say that the systems have become quite mature. The development of these methods in recent years, especially with the help of automatic robust feature-detection algorithms, pushed the boundaries of the state-of-the-art to new frontiers. One can say that structure-from-motion has also reached a mature state and is basically solved for small or medium sized input datasets. The problem of large scale reconstruction is still in active research.

Finally, one essential problem is the integration of the research on reconstruction of the world. Besides the concurring global commercial companies, there is also a slight divergence in the scientific fields. We mean here the parallel research in the computer science disciplines (CG and CV) and photogrammetry and remote sensing. This report tries to contribute to the idea of interdisciplinary by providing a literature review composed of works of all of the mentioned fields.

# 3. Façade Image Acquisition

In this chapter we propose a system for generating high-quality approximated orthographic façade textures based on a set of perspective source photographs taken by a consumer hand-held camera. Our approach is to sample a combined orthographic approximation over the façade-plane from the input photos. In order to avoid kinks and seams which may occur on transitions between different source images, we introduce color adjustment and gradient domain stitching by solving a Poisson equation in real-time. In order to add maximum control on the one hand and easy interaction on the other, we provide several editing interactions allowing for user-guided post-processing.

## 3.1. Introduction

The generation of high-quality façade imagery is a key element of realistic representation of urban environments. Orthographic and rectified façades are also a prerequisite of several structure detection and segmentation algorithms, such as [MZWvG07, MWR*09, MRM*10].

We address the problem of texture generation, which remains a challenging task. Our contribution is a system which provides the ability to create such images from a set of perspective photographs taken by a consumer hand-held camera. The proposed method combines robust automatic processing steps with user interaction. This combination is meant to resolve some remaining weak points of fully automatic attempts and to improve the quality of the output.

## 3.2. Overview

The goal of this work is to provide a convenient and robust way to generate approximations of ortho-rectified images of building façades. The only input we use is a set of photographs of the targeted building taken from the ground using a hand-held, consumer-level camera. These images have to be registered to each other, thus we present a brief overview of multi-view registration and structure-from-motion in Section 3.3.1.

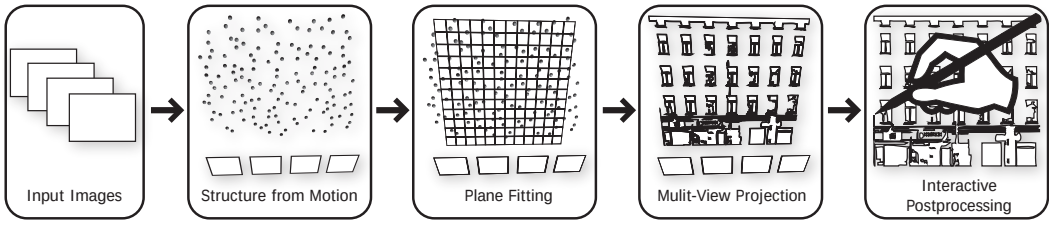


**Figure 3.1.:** *Schematic overview of our system: we compute a sparse point cloud of the scene using structure-from-motion; then, we fit a dominant plane to the point cloud. Next, we project the images of the shots onto the plane and store their colors in a per pixel stack. Finally, we allow the user to brush over the stack in order to remove unwanted content by choosing the best source.*

We expect the object in front of the cameras to be approximately planar, like a single façade, such that it can be substituted by simple geometry, which we call *proxy geometry*. In Section 3.3.2 we propose one possible solution to this problem.

In Section 3.3.3 we describe the details of the multi-view projection method. Our approach is straightforward: we span a grid of desired resolution over the façade-plane. Then, for each pixel in the target resolution we determine which camera shot is optimally projecting onto it, and we collect its color information. At this point two problems arise: The first occurs if two neighboring pixels in the target resolution are filled by color samples from different source images. Usually this results in a visible seam between them. To resolve this we propose color correction and gradient-domain stitching. This is handled in Section 3.3.4. The second problem relates to the actual image content. For some shots we might obtain color samples which belong to external objects that occlude the façade, like vehicles, vegetation, etc. We approach this in a semi-automatic manner in Section 3.3.5 and by turning to user interaction in Section 3.3.6.

Ultimately, the final image is composed according to the automatic and manual corrections in the gradient-domain and an online Poisson solver provides the result (Section 5.3).

## 3.3. Multi-View Ortho-Rectification

In the introduction we mentioned the demand for orthogonal-rectified textures for urban buildings. In this section we present our method for generation of such imagery. We shall introduce the preprocessing steps, like image registration, and explain the approach of multi-view imaging and stitching in the gradient domain.

### 3.3.1. Structure From Motion

In order to gain more information from the images, they need to be registered among each other. The input to this stage are clusters of images retrieved in the previous stage. We resort to the classic sparse stereo *structure-from-motion* (SfM) method to register the images

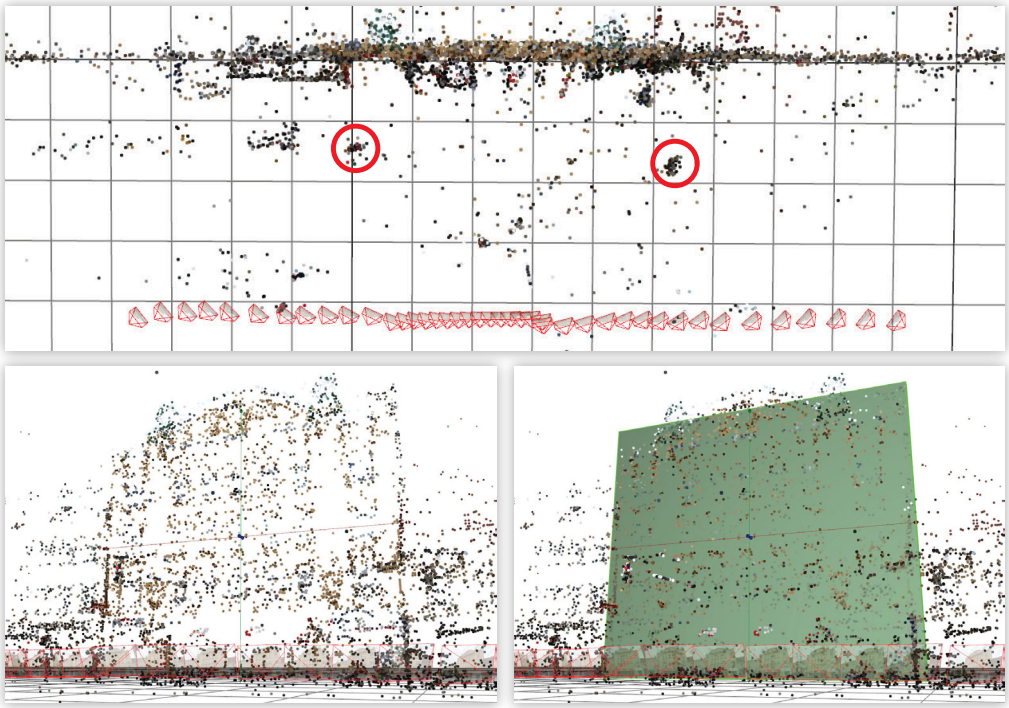

**Figure 3.2.:** *Top: top view on the point cloud computed by the structure-from-motion (SfM) module. The dominant plane is clearly detectable. The circles indicate objects in front of the façade. Bottom left: frontal view of the point-cloud, right: with plane fit into it.*

to one another and to orient and position them in 3d space. This method is based on feature matching, pose estimation, and bundle adjustment [PvGV*04]. Multiple photographs are provided to the modlue and from each one a sparse set of SIFT feature-points is extracted [Low04]. Once multiple images with corresponding features have been established, the extrinsic (i.e., pose in 3d space) properties of their cameras can be determined. Since we are dealing with mostly planar objects, we use a calibrated approach for unstructured photographs, such as the one described by Irschara *et al.* [IZB07]. In accordance with epipolar geometry given known camera parameters, the 3d positions of the corresponding 2d features in the photos can be triangulated, which provides a cloud of 3d space points.

### 3.3.2. Proxy Geometry

**Plane Fitting.** The SfM procedure delivers a sparse point-cloud of the triangulated points in 3d space. If we have not encountered any serious mismatches between the photographs, the points are distributed such that they form a more-or-less coherent planar manifold of the 3d space (cf. Figure 3.2).

For the case that we have well defined geometry and the computed point cloud, we refer to geometry registration method presented in Appendix B. In order to compute the proxy geometry, we introduce here a rudimentary plane detection algorithm based on RANSAC [FB81] for outlier removal followed by least squares fitting. It should be noted, that this algorithm is a simplified version and it can be applied only if we expect the points to lie on only one proxy plane.

Let the set of the 3d points be $\mathbf{X} = \{\mathbf{x}\}_{i=1}^{n}$. In the following, we perform RANSAC on the set such that we obtain only a thin layer of the points $\mathbf{X}^* \subseteq \mathbf{X}$. The "thickness" of the layer is controlled by the distance threshold $\varepsilon$ of the RANSAC procedure. Next, the plane is defined by a 4d vector $\pi$ composed of the normal $\mathbf{n}$ and the distance to the origin $d$. We perform a least squares fit by minimizing the sum of squared distances of all points $\mathbf{x} \in \mathbf{X}^*$ to $\pi$:

$$E_\pi = \sum_i \|\mathbf{n}^T \mathbf{x}_i - d\|^2 \quad \longrightarrow \min \ .$$

We solve this system of equations using a SVD solver. Depending on the accuracy of the computed point-cloud (which depends a great deal on the quality of the camera and the lens) there might be the need for iterative adjustment of the plane. In this case, we repeat the procedure on the $\mathbf{X}^*$ set with a smaller value of $\varepsilon$. This plane serves as projection canvas for further texture projection tasks.

**Façade Boundary.** So far we have a set of registered shots including their camera properties, a sparse point cloud in 3d space and a dominant plane fitted into the cloud. All the previous steps have been computed fully automatically. The only user interaction if any was the selection of proper input images for the SfM procedure.

At this stage there arises the problem of defining the actual façade extent. While there have been attempts to solve such problems automatically, these are error prone and not well defined. On the other hand, this is quite an easy task for a human provided with an appropriate user interface. For this reason, we propose a GUI that allows the user to

- navigate in 3d though the scene,
- look at the scene from the computed shot positions,
- adjust the 3d plane by resizing and rotating it (see Figure 3.3),
- preview the texture by projecting best single-shot image onto the plane,
- finally, also align the whole coordinate system of the scene with the one of the proxy plane,
- and, finally, align the coordinate system of the scene with the one of the proxy plane.

After the adjustment of the façade boundary, the application is ready for the next step: multi-view projective texturing.

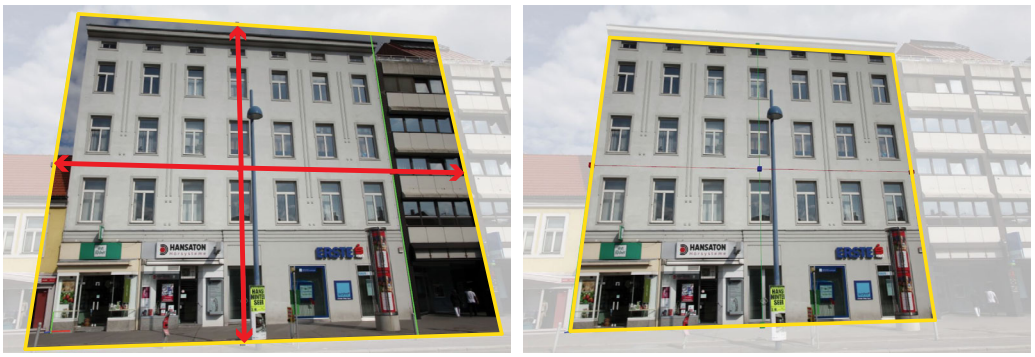

**Figure 3.3.:** *View at the façade plane through one of the projecting cameras. In this view it is easy to adjust the façade-bounds interactively. Left: during the adjustment. Right: final result.*

### 3.3.3. Viewpoint Projection.

The two objectives of our approach are (1) to produce as orthogonal as possible an approximation of the façade image and (2) to work around as many occluders located in front of the façade surface as possible.

In this section we address the issue of sampling as orthogonal an approximation as possible of the façade image and describe the way how the pixels, which project on the image-plane, are chosen.

**Scene Geometry.** First of all we address the rough geometric issues of the multi-view projection. We distinguish different cases of camera placement, where only one is valid and the others are classified as invalid and shots of this class are rejected. Figure 3.4 depicts this issue: the invalid cases occur when the camera is behind the plane ($\mathbf{C}_3$ and $\mathbf{C}_4$) or when it is in the front, but not all four rays from its center through the corners of the frustum intersect the image plane ($\mathbf{C}_1$). The valid case is when the camera is in front of the façade plane and all rays intersect the image plane in a finite distance, such that the projected shape is a finite trapezoid that intersects the façade rectangle (cf. Figure 3.4, left). If not all rays intersect the plane, only a part of the image is finitely projected onto the plane and a part meets the plane at a line at infinity. Even if this case might be considered as partially valid, pixels from such a projection are very strongly elongated along the plane and thus prone to cause sampling artifacts. Since we expect to have enough information from the valid cameras anyway, we simply reject them as invalid ones.

**Shot Selection.** Our approach is based on the fact that we have multiple projective centers along the horizontal axis in world space (since we are using ground-based hand-held cameras). This allows us to compose the target image $I$ in such a way that each pixel is chosen from an optimal camera. As a measure for this optimality, we use an objective

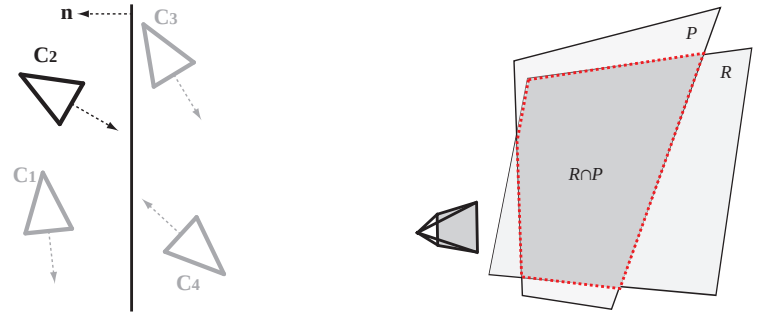


**Figure 3.4.:** *Left: Example of valid ($\mathbf{C}_2$) and invalid cameras in the system. Right: the area of the intersection $R \cap P$ in determines the "quality" of the projecting camera.*

function composed of the camera to plane-normal incidence angle $\varphi$ and a term which expresses the area covered by the footprint of the original pixel projected onto the proxy plane.

From the *law of sines* we know that the length of a projected segment depends on the distance of the camera center to the plane and the projection angle. Figure 3.5, left hand side, depicts this relation, where the length of the segment $\overline{BC}$ depends on the angles $\alpha$, $\varphi_1$, $\varphi_2$ and the length of $\overline{AM}$.

We denote the distance of each camera $\mathbf{c}_k$ to each pixel $\mathbf{p}_i$ as $d_{ik}$, than we approximate the projection area as $A_{ik} = (d_{ik}/d_{max})^{-2}$. We normalize $d_{ik}$ such that it lies between 0 and 1, which is a chosen maximum distance $d_{max}$ (i.e. the most distant camera). For the angular term, we use the dot product of the plane normal and the normalized vector $\mathbf{v}_{ik} = \|\mathbf{c}_k - \mathbf{p}_i\|$, such that: $B_{ik} = \mathbf{n}^T \mathbf{v}_{ik}$. This value is naturally distributed in the range $0 \ldots 1$. Both terms are weighted by the empirical parameters $\lambda_1 + \lambda_2 = 1$, such that the final objective function is:

$$E_I = \sum_i \sum_k \lambda_1 A_{ik} + \lambda_2 B_{ik} \quad \longrightarrow \max , \tag{3.1}$$

where $i$ iterates over all target pixels and $k$ over all valid cameras. We choose $\lambda_2 = 0.7$ in our experiments.

**Image Stacks.** In order to accumulate the projections, we span a grid of desired resolution over the detected and bounded façade plane. Then, for each pixel in the target resolution, we determine a set of cameras which project optimally according to the aforementioned constraints. We store this values in a volume of the size *width* $\times$ *height* $\times$ *number of shots* attached to the proxy, which we call *image stack* due to its layered nature. Left hand side of Figure 3.5 shows a schematic, 2d top view of this idea. Image stacks have been demonstrated to be a very effective structure in the work of Agarwala *et al.* [ADA*04], where they were used for interactive photomontage.

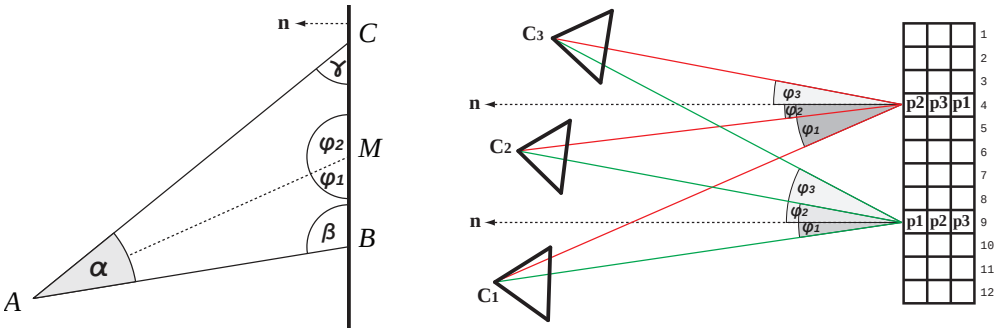


**Figure 3.5.:** *Left: The relations of the projection, where the length of $\overline{BC}$ only depends on the angles $\alpha$, $\varphi_1$, $\varphi_2$ and the length of $\overline{AM}$. Right: Projection from the shots onto the image stack. For each pixel indicated by the numbers on the right, the best cameras are chosen, and the projected value is stored in the respective layer of the stack.*

### 3.3.4. Seamless Stitching

The result of the algorithm presented in the previous section is already an approximation of an orthogonal façade image. One remaining problem are the visible seams along transitions between pixels from different sources, which we address by a gradient-domain stitching algorithm.

**GPU Poisson Solver.** In Section 2.2.2 we have mentioned the idea of Poisson image editing, which where presented in [PGB03]. The beauty of this method manifests itself in both the elegance of its formulation and the practical results. It is based on the insight that one can stitch the derivatives of two signals instead the signals themselves. The derivative functions have the advantage that the intensity differences between them are relative, and not absolute as in the original signals. Thus, any differences in the amplitude of the original signals vanish in their gradient fields. We can compute them in the discrete case of an image $I$ as forward differences:

$$\frac{\partial I}{\partial x} = I_{(x+1,y)} - I_{(x,y)} \tag{3.2}$$

$$\frac{\partial I}{\partial y} = I_{(x,y+1)} - I_{(x,y)} \, . \tag{3.3}$$

After editing (e.g., deleting, amplifying) and combining (e.g., blending, averaging) of the derivatives of one or more images, one obtains a modified gradient field $G = [G_x \, G_y]^T$. Unfortunately, this is usually a non-integrable vector field, since its curl is not equal to zero, and thus one cannot reconstruct the original signal by a trivial summation. This problem

is addressed by solving for the best approximation of the primitive (original) signal by minimizing the following sum of squared differences:

$$\begin{aligned} E_U &= \|\nabla U - G\|^2 \\ &= \left(\frac{\partial U}{\partial x} - G_x\right)^2 + \left(\frac{\partial U}{\partial y} - G_x\right)^2 \quad \longrightarrow \min. \end{aligned} \tag{3.4}$$

In other words, we are looking for a new image $U$, whose gradient field $\nabla U$ is closest to $G$ in the least squares sense. This can be formulated as a Poisson equation:

$$\nabla^2 U = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y},$$

which results in a sparse system of linear equations that can be solved using least squares. Since we strive for real-time performance, we adapt a GPU solver proposed by [MP08], which is a multi-grid solution [AR07]. It performs at real-time rates with up to four mega pixel images (on an NVIDIA GeForce GTX 285), which allows not only for the stitching of precomputed layers but also interactive editing of the layers. We elaborate this in Section 3.3.6.

**Stitching.** For the mentioned multi-view approach we combine the pieces from different images in the gradient domain for the entire façade image, and then we solve the Poisson equation with Neuman boundary conditions. This means that we do not define any borders around the façade, but fill the initial values with zeros [AR07].

**Color Correction.** Despite the fact that we are using a Poisson image editing approach, we perform a simple color correction procedure before the actual stitching process. This provides better initial values and has turned out to be useful in cases where we have slight transition in the illumination of the façade. In practice this happens very often, since the global illumination (sun, clouds) changes. We resort to a simple approach presented by Reinhard *et al.* [RAGS01], where we just shift the mean $\mu$ and the standard deviation $\sigma$ of all images in the stack to common values. (there are more sophisticated methods in recent literature, like [SS10], but in our case we do not see any significant advantage of such approaches).

Unlike their method, we perform the linear shift in the RGB color space, since we do not aim for an appearance change but just for slight color correction. Thus, for each pixel we shift each color channel to zero-mean and scale all points by the factor given by the ratio of the standard deviation of the actual shot to the key-image, followed by a back-translation to the key-image mean:

$$c_{out} = \frac{\sigma_{key}}{\sigma_{in}} (c_{in} - \mu_{in}) + \mu_{key},$$

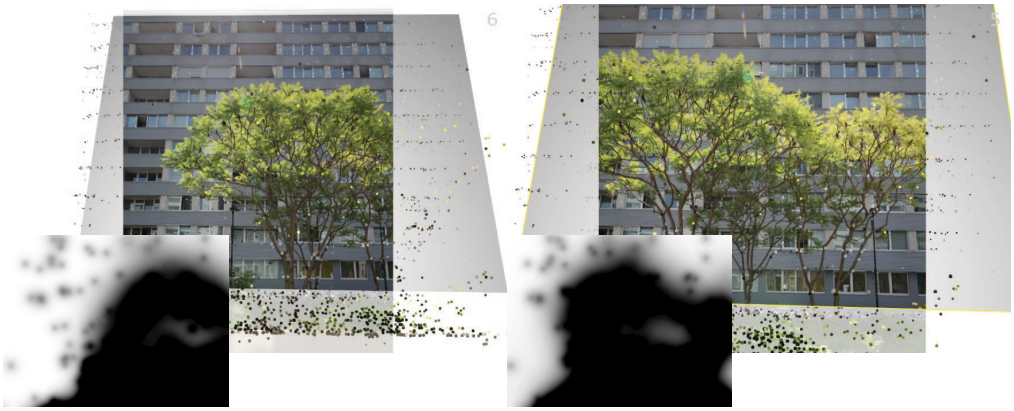

**Figure 3.6.:** *Occlusion masks of two shots generated by splatting the 3d points onto the proxy plane. Shots are looking at the proxy, the overlayed masks are in proxy-plane space. The final result of this scene is shown in Figure 3.12.*

where $c$ stands for each color channel separately. The key-values are chosen from an input shot with the largest projected area on the bounded façade plane. In fact, since we are interested in an equality of colors in the stack, the choice of the reference color is not of a very big importance. Also other heuristics are conceivable, such as an averaging over all shots or just taking the first shot.

### 3.3.5. Occlusion Handling

The described multi-view projection delivers optimal color samples for the ortho-façade pixels as long as the proxy geometry of the scene is visible from the cameras. However, in real-life data we usually encounter a number of obstacles between the camera and the façade: pedestrians, street signs, vehicles, vegetation, etc. These, if projected on the plane provide unwanted and disturbing artifacts. To counter this, we introduce two ways to integrate the occlusion into the scene.

**Point-Footprint Projection.** The first idea is based on the observation that many 3d points of the SfM point cloud do not belong to the proxy, but to other objects in front of the camera (see Figure 3.2, top, red circles). Hence, they represent potential obstacles and we splat these points onto the image-plane, such that their footprints provide an additional visibility term $V_{ik}$ to the source-selection function presented in Equation 3.1:

$$E_I = \sum_i \sum_k (\lambda_1 A_{ik} + \lambda_2 B_{ik}) \cdot V_{ik} \quad \longrightarrow \max , \tag{3.5}$$

In our implementation, we introduce the $V_{ik}$ term as a per-shot mask, which contains per-pixel visibility information from the splatted 3d points (shown in Figure 3.6). According to

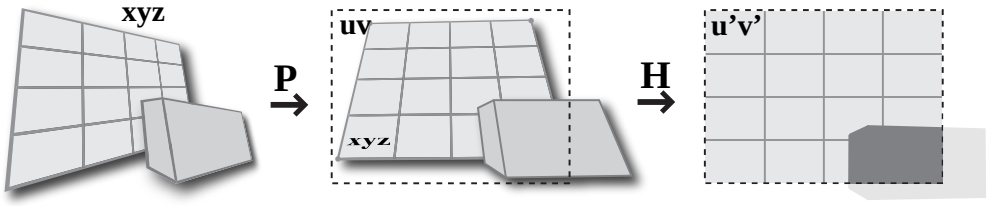


**Figure 3.7.:** *Left: projection of the 3d scene by a shot-camera* $\mathbf{P}_k$*. Note the occluder in front. Middle: We compute a homography* $\mathbf{H}_k$ *of the façade-plane to the view-port. Right: in the vertex shader the scene is transformed by the shot view projection* $\mathbf{P}_k$ *and* $\mathbf{H}_k$*.*

this value, a shot might be considered as an occluded one, even if its score from Equation 3.1 is high.

**Geometric Occluders.** One further way to include the occluding objects into the scene is to explicitly model their geometry. We do so by allowing the user to model bigger objects roughly by primitive shapes such as cuboids. An example is shown in Figure 3.11, where a shop in front of the façade has been approximated by a 3d box and entirely removed. We add this information in the same manner as with the 3d points above. However, we assign the modeled occluder maximum confidence value.

**Implementation.** We implement the occlusion test in hardware. Let us denote the shot-camera projection by $\mathbf{P}_k$. For each shot we compute the homography $\mathbf{H}_k$ that maps the façade proxy projected by $\mathbf{P}_k$ to the target image space. In the vertex shader we transform the entire scene by $\mathbf{P_k}$ and $\mathbf{H_k}$, such that we obtain the result in the target resolution (see Figure 3.7). In the pixel shader, the interpolated depth of the projection of the scene is tested with the proxy plane. In a second pass, 3d points in front of the proxy are splatted by the same mapping as above onto the target. The radius of their footprints depends on the distance to the target and is weighted using a radial falloff-kernel (see Figure 3.12). The results are accumulated in a per shot mask, which acts as the occlusion term $V_{ik}$ in Equation 3.5.

### 3.3.6. User Interaction

Finally, our system allows the user to directly edit on the projected façade image. To accomplish this we introduce several brushing-modi which can be applied locally and precisely in order to repair small details. The brush operations exploit the fact that we have multiple information per pixel stored in the image stack. On the top of the stack (and thus visible) lies the color taken from the camera that best maximizes Equation 3.5. However, neither the automatic, 3d point footprint method, nor the interactive geometry modeling method presented above ensure the removal of all outliers. With the help of interactive

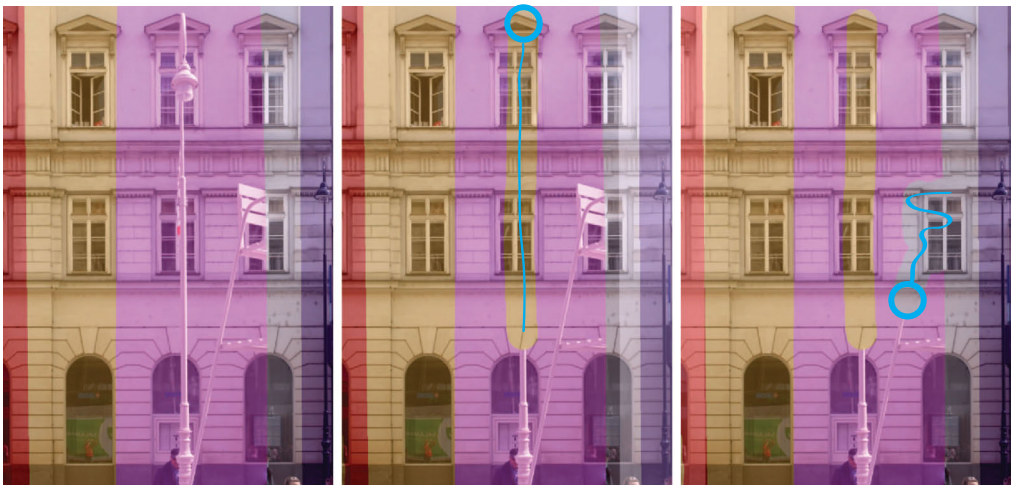

**Figure 3.8.:** *Interactive brushing. Left: color stripes indicate regions stemming from different cameras. Middle: the eraser brush brings the yellow layer to the front (over the purple). Right: the growing brush pulls the gray layer over the purple one. Blue storks indicate the user actions.*

brushing in the gradient domain, our system provides the user convenient editing tools to control the final result. The following brushes relax the results provided by Equation 3.5 and change the order in the stack.

**Growing Brush.** This brush is thought to "grow" a region projected from one image over an other region. It captures the shot where the user starts to brush (by clicking). While holding the mouse button down, the captured shot is propagated interactively to others. As a visual aid, the user can overlay the multi-view image with a colored indication layer, such that regions stemming from different sources are highlighted by different colors, as shown in Figure 3.8.

**Eraser Brush.** The idea behind this brush is to use pixel samples lying behind the visible stack layer. Each time the user clicks, the next layer is chosen and its information can be brushed on the top of the stack. If the last layer is active, it rotates on click over the stack modulo the number of layers. In this way it is possible to bring information from another cameras to the front by just clicking on one position. Since other shots have a different viewpoint, they often do not contain the potential occluder on the same pixels, but shifted due to the parallax. In other words, this brush brings the next layer information at current mouse position to the front and gives the user a simple way to switch between the layers (Figure 3.8).

## 3.4. Results

The table on the right shows timings of the system with 22 input images (8 MP each) measured at two target resolutions (Intel Quad Core with NVIDIA GeForce GTX 285). Brushing runs on the same data set at approx. 40 fps. In Figures 6.1, 3.9, 3.11 and 3.12 we present visual results of our system. Additionally, we refer to the accompanying video material. We usually work with a target resolution of 2 mega pixels, mainly due to hardware limitations. However, since our system allows the user to freely define the extent of the projected façade, it is easily possible to focus only on selected parts and apply the maximum resolution to this subregions only. This "zoom" is of course limited by the source resolution, which can have up to 16 mega pixels on current hardware with DX9.

| operation | 2 MP | 3 MP |
|---|---|---|
| accumul. | 0.05s | 0.06s |
| color corr. | 6.0s | 8.0s |
| sampling | 9.0s | 11.5s |

**Limitations.** Our method fails in cases, where in all input images the actual façade is occluded. In such cases we want to resort to methods that utilize similarity present in the image. A problem of our current implementation is the limitation of the stack to four layers due to hardware-API constraints (DX9). We plan to switch to DX10 to resolve this. Finally, our method is quite hardware intensive, such that it requires graphics cards with 1GB video RAM to perform well.

**Future Work.** We are considering to extend the system in a way that allows the user to operate in moderate resolutions for real-time interaction while calculating higher resolutions offline. Furthermore, we want to extend the geometry modeling part of the solution.

**Figure 3.9.:** *A close-up of the image shown in Figure 3.11. Pedestrians and their reflections visible in the left image have been removed (middle). The colored masks indicating source shots are shown in the overlayed image.*

## 3.5. Conclusions

We present a system for generating approximately orthographic façade textures. We pay particular attention to high-quality, high-resolution and obstacle-free images. Most steps of our method are fully automatic: image registration, pose estimation, plane fitting as well as per-pixel projection. On the other hand, some tasks have proven difficult to solve automatically with adequate quality. For these cases we introduce interactive tools. For the problem of bounding the actual façade, we provide the user with an easy method to define the extent. Another difficult problem is the detection and removal of possible occluders in front of the façades. To solve this, we propose two approaches: projection of SfM outliers and modeling of additional geometry. The major contribution of our system is the detailed removal of occluders by exploiting the multi-view information. Our system is intended to serve as part of a complex urban reconstruction pipeline.


**Figure 3.10.:** *Steps of the multi-view image generation system. Top-left: one of typical perspective input photographs, please note the occlusion. Top-right: the result of the proposed ortho-image generation method (note the pedestrians). The second row shows masks indicating source images of the composition by colors: automatic result (left) and interactively post-processed (right). Bottom: the final result after interactive post-processing.*


**Figure 3.11.:** *Top left: multi-view stitching without constraints. Top right: multi-view stitching with geometry constraints. Bottom from left to right: one of the original perspective shots, occluding geometry has been modeled into the scene, source-indication masks without and including the geometry occlusion.*

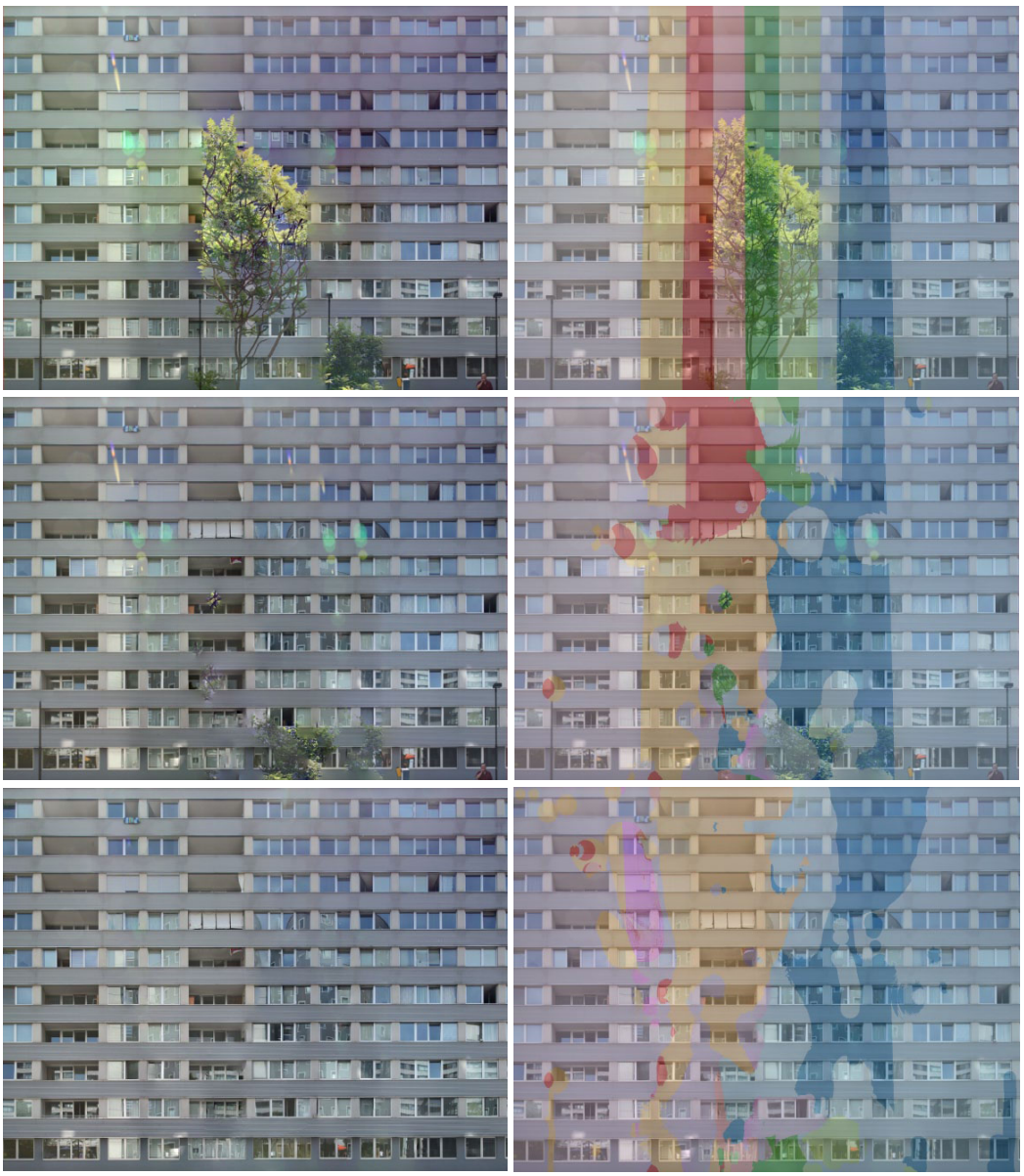

**Figure 3.12.:** *Automatic removal of occluding objects by utilizing the information from structure-from-motion points. Left: image and its mask after multi-view stitching without the occlusion term. Middle: results with occlusion term. Right: result with occlusion term post-processed by interactive brushing. Note that lens flares have been removed as well.*

# 4. Façade Image Segmentation by Similarity Voting

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

## 4.1. Introduction

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

To achieve this, we propose a Monte Carlo sampling scheme which operates only on selected image features. While these can be defined by different means, we decided to use simple *Harris Corners* [HS88] due to their robust and fast computation. For measuring similarity between particular features we use a multiresolution version of common operators, such as *Normalized Cross Correlation Coefficient*.

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


**Figure 4.1.:** *The red lines indicate the grid, which has been detected on the façade. The proposed algorithm is robust to obstacles such as different illumination or reflections in the windows (best seen in color).*

## 4.2. Overview of the Approach

The algorithm takes as input a single orthogonalized view of a façade. The output is an orthogonal grid that defines a segmentation of the façade 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 façade image (e.g., floors or windows) it is necessary to search for similar image regions. This is done by comparing small image regions on multiple resolutions of the image for similarity. Because comparing every pair of potentially corresponding image regions is computationally prohibitive,a Monte Carlo importance sampling strategy is applied to collect statistical evidence about any translational similarities. To extract these relevant patterns out of all the collected evidence the representative offsets are sorted into a histogram where large patterns result in large peaks. These are then extracted by Mean Shift clustering [CM02]. The result of this stage are offsets in pixels that relate directly to the prevailing repetitive patterns in the image.

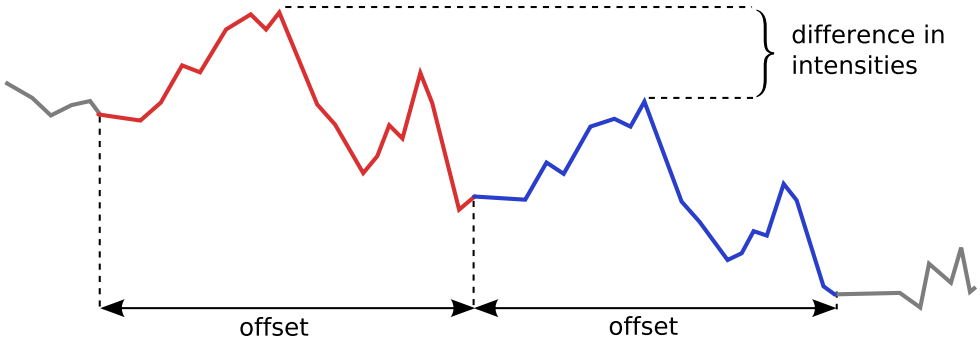


**Figure 4.2.:** *Example of a repetitive pattern in 1D with a highly similar but not identical instance. Relative differences in signal intensities between instances of the pattern should not influence the detection algorithm. An appropriate similarity measure must be applied that is insensitive to the overall intensity level of the region.*

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

## 4.3. Search for Dominant Repetitive Patterns

A closer look at the typical structure of facades helps to understand which image patterns are relevant for window detection. Most facades feature many windows of the same size and similar appearance. The arrangement of windows is almost always the same for the floors of the same façade. Common exceptions to this rule are usually the first floors which are irregular or different from the others. If we consider a sequence of axis-aligned pixels as a function of the intensities, we notice certain regular repetitions in the signal (Fig. 4.2). These repetitions are coherent over multiple adjacent pixel lines of the image.

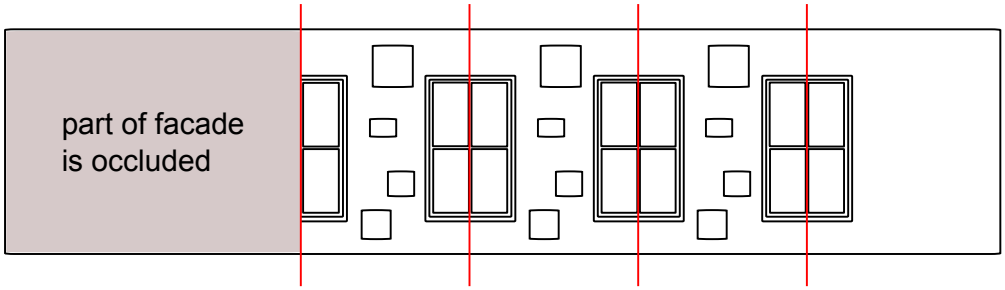


**Figure 4.3.:** *Without a priori knowledge about the signal, it is not possible to evaluate the correctness of a split. In this case half of the first window has been occluded, causing a shift in the start of the repetitive pattern.*

**A Repetitive Pattern.** On a spatial signal it 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. We call this a *repetitive instance* (see Figure 4.2). The same image features that are very important for human vision such as edges and corners are most important for our repetitive pattern detection algorithm.

To define the border of a repetitive pattern we assume that the pattern begins at the first distinctive feature (i.e., edge) that is similar to the signal at the characteristic offset and ends as soon as the signal starts to differ too much from the original instance. We constrain the input images to complete pictures of a façade, such that it is impossible (except in case of occlusions like in Figure 4.3) for a pattern to start in the middle of a window. The bounds of a repetitive pattern are not sharp and have to be defined by a similarity threshold. With such a threshold, non-repetitive regions can be distinguished from pattern regions.

A difficult problem for image segmentation based on repetitive patterns is the handling of overlapping patterns. To demonstrate the problem, consider the façade image in Figure 4.4. There are two concurring segmentations based on either the one pattern's offset or the other's. A solution to this problem, which is adopted in this approach, is to exclude some of the detected patterns according to a priori knowledge or image area constraints.

### 4.3.1. Similarity Measure

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

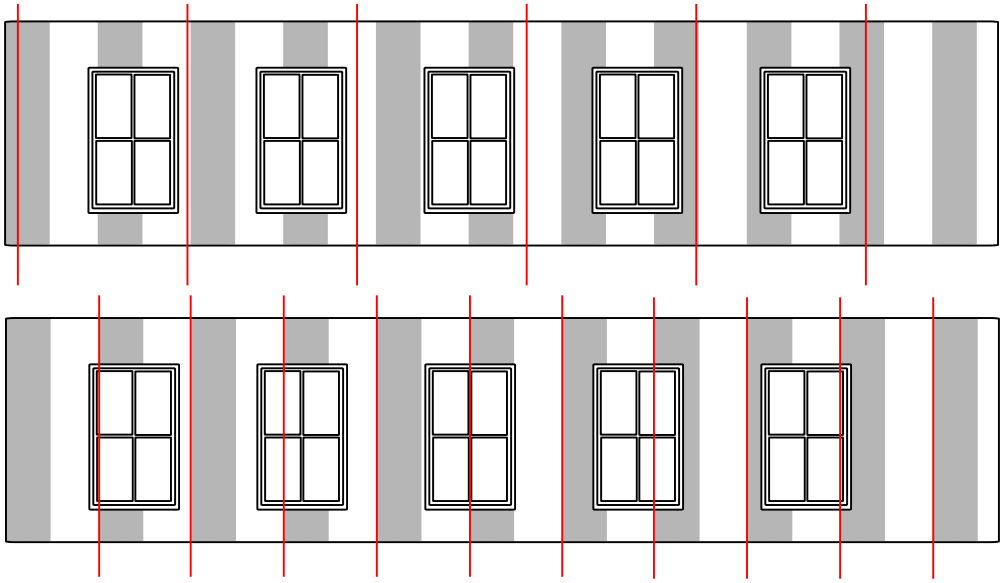

**Figure 4.4.:** *Two overlapping repetitive patterns and their corresponding splitting lines. There are often overlapping patterns, especially in Art Nouveau facades that feature a great deal of decor.*

$$\|\mathbf{x},\mathbf{y}\|_{\text{ncc}} = \frac{(\mathbf{x}-\bar{\mathbf{x}})^T(\mathbf{y}-\bar{\mathbf{y}})}{\|\mathbf{x}-\bar{\mathbf{x}}\|\|\mathbf{y}-\bar{\mathbf{y}}\|}\,. \tag{4.1}$$

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

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

**Multi-Resolution Similarity.** When measuring patterns, the size of the pattern relative to the size of the measurement window is very important. If it is too small or too large compared to the measurement window, one will obtain ambiguous results (Fig. 4.5). Rather than increasing the patch size to improve the robustness of the measure, a very efficient way is to combine the results of measurements on different scale levels of an image pyramid. This idea has been successfully used in many texture synthesis algorithms. It is computed

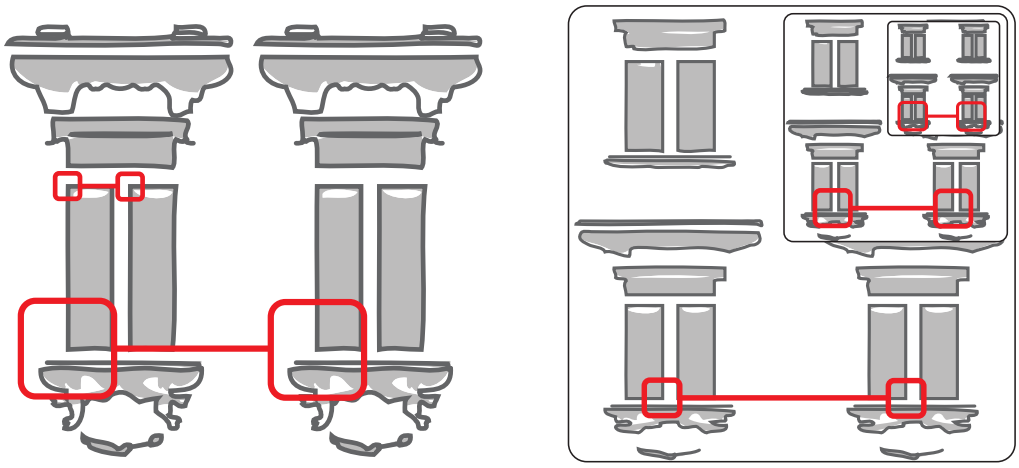

**Figure 4.5.:** *Left: depending on the window size, features can become ambiguous. Two differently sized similarity windows with highly similar matches. Right: by comparing with the same patch size on different resolutions, the ambiguity can be resolved.*

by subsequently scaling the image with the factor $s$ (in our case we use $s = \frac{1}{2}$ and cubic down-sampling):

$$\varsigma(\mathbf{x},\mathbf{y}) = \frac{1}{N_s}\sum_{k}^{N_s}\|\mathbf{x}_k,\mathbf{y}_k\|_{\text{ncc}}\ , \tag{4.2}$$

where $N_s$ is the number of scales and $\|\cdot\|_{\text{ncc}}$ operates on the $k$-th scale of the input image $I$. The similarity $\varsigma$ results from all scale levels that 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. The window size is kept constant, as shown in Figure 4.5 left hand side. In our empirical tests, we determined that a good tradeoff between speed and robustness is a size of $15 \times 15$ pixels on 3 pyramid levels. This operator on the pyramid is not completely equivalent to the multi-sized similarity operator on the original image because it introduces implicit low-pass filtering by down sampling. Even though it is very robust with respect to real-world noise while being relatively fast compared to using large similarity windows on the original image.

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*. An *early break* stops the evaluation of all pyramid layers if the similarity value on the higher

pyramid level is below a certain threshold, since in practice most of the compared regions are not at all similar.

Finally, it is practically independent of the size of the input images and the size of the patterns. By using a constant window size the multi-resolution similarity operator on the image pyramid is highly efficient compared to using large similarity windows on the original image.

### 4.3.2. Monte Carlo Sampling

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

Façade elements such as windows, balconies, etc., are characterized by sharp orthogonal edges and corners. Based on this information we implement an importance sampling strategy. It is not so important to sample image regions without any salient features because they might not contain any façade elements. Instead we focus on edges and corners which are better indicators of façade elements. The implementation of such an edge-based importance sampling strategy is quite straightforward: an edge image is computed using Sobel-filtering and Canny edge detection [Can86]. Using this sampling strategy, the accuracy of the result is significantly higher than for simple uniformly distributed random position sampling of the image, while requiring significantly less samples.

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

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

$$h(\Delta) = \sum_{i}^{N} \begin{cases} 1 & \text{if } \varsigma(p_i, p_\Delta) > t \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

This function counts how many samples (random pairs of points) with a given offset $\Delta$ have a multi-resolution similarity value greater than a fixed threshold $t$. We have determined em-

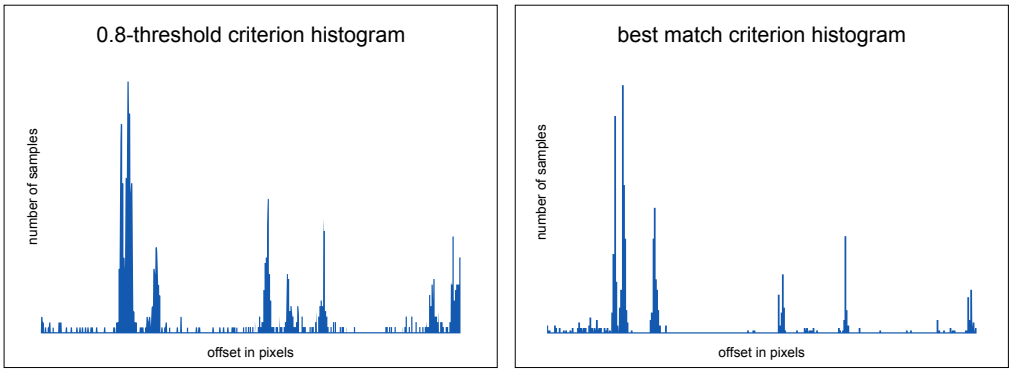


**Figure 4.6.:** *Comparison of histograms resulting from 100k samples with threshold criterion selection (left) and 1k samples with best match criterion selection (right). The broad peaks in the left hand histogram and high peaks of irrelevant offset combinations are signs of the much higher overall error of the simple threshold criterion.*

pirically that the threshold of 0.8 of normalized cross-correlation operator ensures that only highly similar matches are counted. By counting only samples with very high similarity values the variance of the estimated distribution of offsets is significantly lower. However, a quality criterion with a single fixed threshold still counts many imprecise matches because the sampled offsets are not compared to each other in any way. Even significant deviations from the perfect match of two regions may feature insignificantly high similarity values which might be much higher than the threshold. The problem arising from this fact is, that the results are noisy and the significant offsets may be hard to distinguish from the rest (see Fig. 4.6).

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

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

$$h(\Delta) = \sum_{i}^{N} \begin{cases} 1 & \text{if } \Delta = \arg\max_{\Delta_j} \varsigma\left(p_i, p_{\Delta_j}\right), \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

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

To sample according to the *best match criterion* means to count how many times a given offset $\Delta_j$ 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

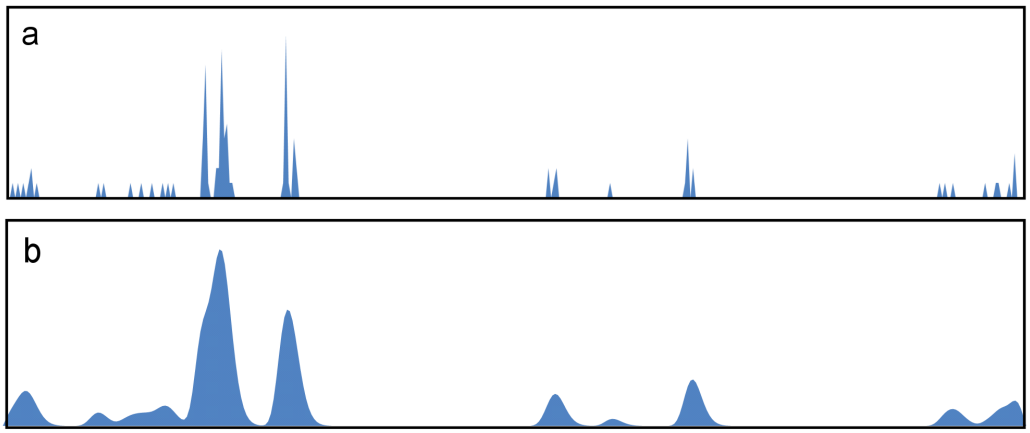


**Figure 4.7.:** *Original histogram (a) and a smoothed and normalized histogram (b). In the smoothed histogram some close peaks are merged together because of oversmoothing. This reduces the number of concurring extracted peak locations on the one hand but also degrades precision of the segmentation on the other hand.*

of hits represents a pattern that is more dominant in terms of recurrence similarity and was found on a large image area.

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

In this context it is also important to mention the optimal size of the filter kernel. While for small images up to one megapixels a 3-pixel filter kernel is sufficient it is certainly not adequate for a 10 megapixel image because it can no longer remove the large-scale noise. An optimal filter kernel size must therefore be derived from the size of the input image in order to adapt the filter kernel to the optimal relative size. In the reference implementation a filter kernel size of $n = \frac{d}{50}$ proved to be useful for most images, where $d$ is the current image dimension (width or height), depending of the processing direction. Finally, the peaks are obtained by *mean shift* clustering [CM02].

**Post-processing of Extracted Offsets.** In many cases the extracted offsets include doubles, triples and higher multiples of the smallest offset to the first recurrence. If a pattern is not uniformly spaced throughout the image, which means that there are differently sized intervals between the re-occurring regions, it might as well happen that the extracted offsets contain combinations of those different offsets (see the annotations in Figure 4.8 for

examples of multiples in a façade image). A simple but efficient solution to this problem is to remove all offsets that are close to integer multiples of the smallest offsets.

## 4.4. Localization and Segmentation

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

### 4.4.1. The Similarity Curve

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

A horizontal similarity curve $S(x,\Delta)$ for an offset $\Delta$ is defined as follows: the image is sampled at every pixel column $x$ at $N$ random locations $y_i$. The mean over every pixel row is the value of the similarity curve at pixel column $x$ (see Figure 4.9):

$$S(x,\Delta) = \frac{1}{N}\sum_{i}^{N} \varsigma\left(p(x,y_i), p_\Delta(x,y_i)\right). \tag{4.5}$$

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

The localization of the patterns is done by comparing the similarity curves for each relevant offset against each other (see Fig. 4.9 top). 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.

### 4.4.2. Segmentation

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


**Figure 4.8.:** *Demonstration of a number of possible multiples of offsets A and B which might obscure the results of the histogram extraction. Two, three and four times multiples of an offset happen quite often and can be easily removed by postprocessing.*

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 $L_i$:

$$L_{i+1} = L_i + \arg\max_{\Delta} \left( \frac{\sum_{x=L_i}^{L_i+\Delta_j} S(x,\Delta_j)}{\Delta_j} \right), \tag{4.6}$$

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

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

A shortcoming of this segmentation method is the fact that an offset $\Delta$ and its non-fractional multiple $N\Delta$, with $N = 2,3,4,\ldots$, 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.

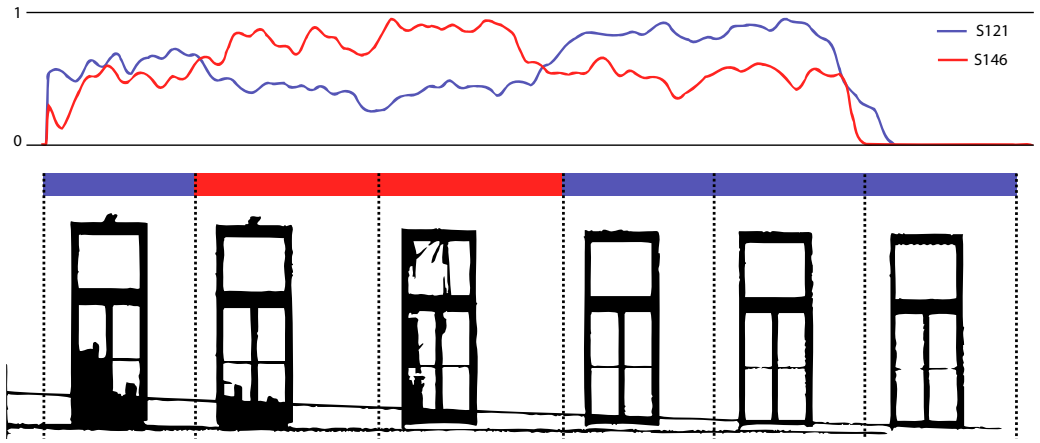


**Figure 4.9.:** *Illustration of the iterative segmentation algorithm. For each iteration and each major offset an integral $F_i$ of the similarity curve $S_i$ is calculated. Since the integration is over a different range for every offset, the resulting areas are normalized to allow a comparison. The 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.*

Their similarity is quite equal yielding unstable results depending on the random numbers used for sampling. A possible solution is to modify the splitting function in order to slightly prioritize smaller offsets over larger ones with a weighting factor:

$$\omega(\Delta_j) = 1 - \left( \frac{\Delta_j}{\min \Delta} \varepsilon \right), \tag{4.7}$$

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

$$L_{i+1} = L_i + \arg\max_{\Delta} \left( \frac{\sum_{x=L_i}^{L_i+\Delta_j} S(x,\Delta_j)}{\Delta_j} \omega(\Delta_j) \right). \tag{4.8}$$

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

## 4.5. Results

In this section various aspects of the proposed façade 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.

### 4.5.1. Performance

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

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

**Figure 4.10.:** *Running time comparison.*

**Best-match vs. Threshold criterion.** The table in Figure 4.10 summarizes horizontal segmentation performance of a façade image with different resolutions using threshold sampling criterion with a threshold of 0.8 and 50.000 samples. The performance of vertical segmentation is equivalent to horizontal segmentation.

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

**Impact of the probe size.** The performance of this segmentation method is not only dependent on the image size but also on the number of samples taken. Table 4.1 shows the horizontal segmentation performance of a typical façade image with a resolution of 0.4 megapixels and different numbers of samples. For the threshold sampling criterion, a threshold of 0.8 was used.

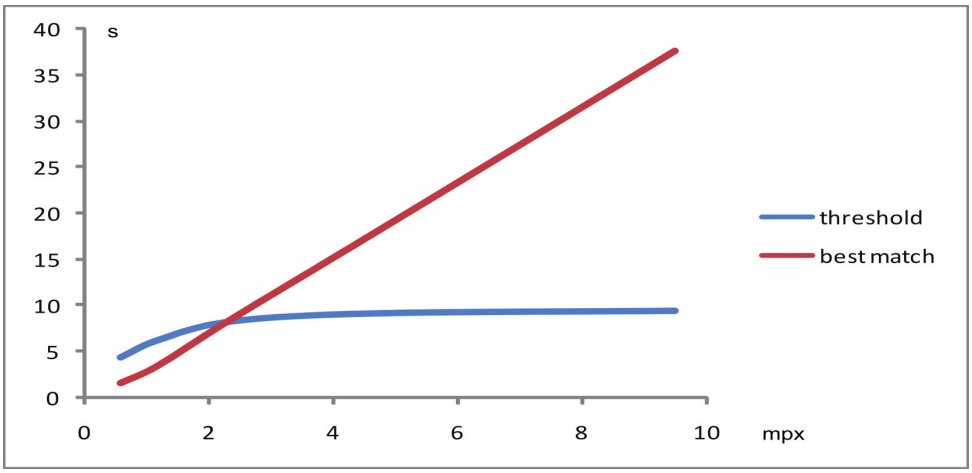


**Figure 4.11.:** *Performance comparison of the sampling criteria "best match" versus "threshold". The graph displays the running time of each sampling strategy as a function of image size.*

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

### 4.5.2. Quality

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

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

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

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

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

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

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

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

Obviously the two different sampling methods behave completely different with random noise applied to the input images. The best-match sampling criterion is extremely robust and is even under heavy interference with random noise able to find the regular pattern beneath. Threshold sampling, on the other hand, is quite fragile with noisy images. This is due to the fixed similarity threshold criterion, which must be fulfilled for each sample in order to be stored in the histogram. In order to perform well with degrading image quality and noise, this threshold would need to be adapted dynamically. This would be a possible subject of further improvement. In Figure 4.13, bottom row we demonstrate the robustness of the segmentation algorithm on a real-world image – the algorithm reliably detects the repetitive pattern even though it is heavily obscured by blur and irregular vegetation.

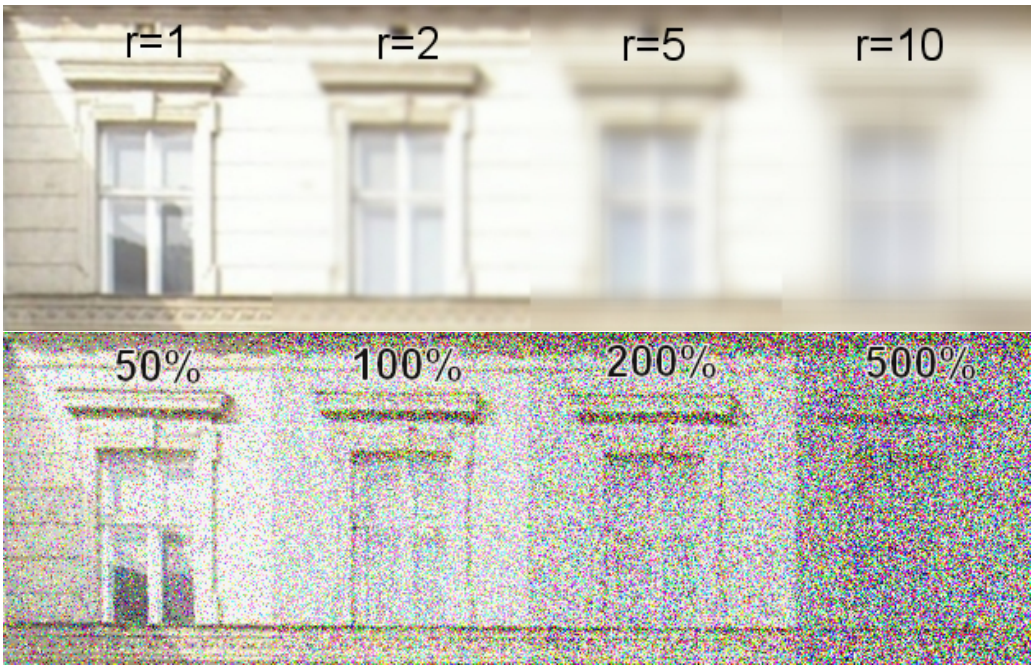


**Figure 4.12.:** *The test images: under Gaussian blur with different radii (top) and under increasing levels of random noise (bottom).*

## 4.6. Conclusions

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

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


**Figure 4.13.:** *Results. The red lines indicate the grid that has been automatically detected on each façade (best seen in color).*

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

**Table 4.1.:** *Probe size dependence.*

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

**Table 4.2.:** *Precision. See description in the text.*

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

**Table 4.3.:** *Robustness to Gaussian blur.*

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

**Table 4.4.:** *Robustness to random noise.*

# 5. Façade Image Segmentation by Clustering

In this chapter we introduce a novel, data driven method to infer distributions of rectilinear grids over a simple, orthographic-rectified façade image inspired by unsupervised learning methods like data clustering. The resulting tilings can be arranged hierarchically and serve as a starting point for a interactive modeling process.

## 5.1. Introduction

In this chapter we introduce a novel approach for deriving structure of building façades directly from one approximately orthogonal image*. Our method is based on the observation that most architectural objects and its sub elements, such as windows, doors, quoins, ledges, pilasters, etc. have a simple structure favoring the rectangular shape. Thus, we preprocess the input façade photographs to be well aligned and introduce a novel way of interpreting them: we treat them directly as data matrices. Figure 5.1 illustrates this idea, such that an image can be seen either as rows or as a columns of data points.

Looking at a façade from this point of view allows us to apply algorithms designed for analysis and mining of high-dimensional data such as cluster analysis and dimensionality reductions [HTF09]. This analysis provides us the global information about the distribution of the structure of the façade and gives us the clue for further processing in finer steps. In particular, after clustering and segmentation of the rows or columns we obtain the distributions of the floors or windows. We combine the results of the horizontal and vertical directions and obtain a façade decomposition into a rectilinear grid. This representation can also be seen as planar tiling (see Figure 5.3).

## 5.2. Façade Approximation

As mentioned, we treat rectified façade photographs directly as matrices. In particular, we take the input façade of the size $m \times n$, where $n =$ number of columns and $m =$ number of

*In Chapter 3 we propose a method for how to obtains such an image from a set of photographs. Alternatively, in Appendix A we show a simple single-view approach to this problem.

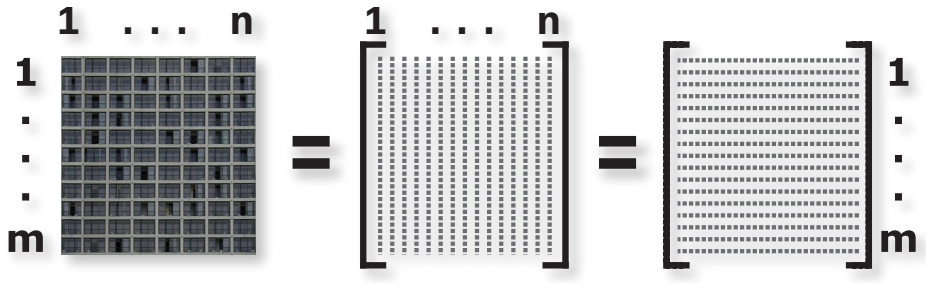


**Figure 5.1.:** *Façade images can be interpreted directly as matrices. Those contain either n-column vectors or m-row vectors.*

rows of pixels. We consider the rows and the columns of the image as vectors, such that the columns are $n-$vectors $\in \mathbb{R}^m$ space and rows are $m-$vectors $\in \mathbb{R}^n$ space:

$$\mathbf{X} = \left[\mathbf{c}_1 \dots \mathbf{c}_n\right]_{m\times n} = \begin{bmatrix} \mathbf{r}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{r}_m^{\mathrm{T}} \end{bmatrix}_{m\times n} .$$

This idea can be easily extended to color-images, where the dimensionality of each space grows by the factor of 3 and we concatenate the color-elements to one vector. In the case we want to represent the façade-data in an other color space (e.g. RGB, LUV, YCC, XYZ), we can first convert the image to the desired color space and then create the vectors. This is due to that the ordering of the data is basically independent of the color space. In the following we will refer to the actual data-vectors (feature-vectors, data-points, variables or observations) as row-vectors $\mathbf{x}_i$ of the data matrix $\mathbf{X}$ independently of the actual chosen façade orientation. The actual elements $x_{ji}$ of the variables will be referred to as attributes (properties, dimensions).

### 5.2.1. Preprocessing

**Gradient Suppression.** The façade image's illumination gradient can adversely affect the quality of the vectors used in clustering. In order to suppress it, we carry out a technique from mathematical morphology called the white top hat function $h(I)$ [Mey78]. Given a grayscale image, the white top hat transformation is performed by subtracting the opened image from the original: $h(I) = I - \gamma(I)$. The morphological opening of an image emphasizes the dimmer areas of the image. Subtracting the opened image from the original amounts to favoring the reduction in intensity of brighter areas in the original. For color images we carry out the white top hat separately on each of the three image channels. Since the local intensities of the opened image are close to those of the original, simply subtracting the opened image from the original can lead to contrast reduction arising from

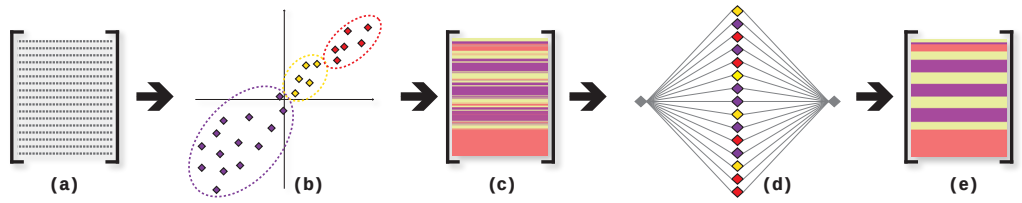


**Figure 5.2.:** *Overview of the clustering procedure: (a) the input data* $\mathbf{X}$ *is clustered by either (b) K-means or more sophisticated spectral clustering methods (refer to Section 5.2.2). The result of this stage (c) is then post-processed by 1d graph-cut segmentation algorithm (d), which minimizes the "length" of the boundary between the clusters. This results in removing of isolated points in the clusters (e).*

inadequate numerical precision. Instead, We subtract only half of the per-pixel intensity of the opened image from the original and then add back the average intensity of the halved opened image to the difference image. Moreover, we found that carrying out a light Gaussian blur ($\sigma = 9$) on the opened image (opened using a $7 \times 7$ square structuring element) results in better looking results than by relying on morphological opening alone. While component-wise operations such as this one can alter the color balance of the image, we have found that gradient suppression leads to better performance in the clustering step.

**Dissimilarity.** In order to measure the distance of particular feature vectors we need to define a metric that will allow us to compare particular features. In fact, specifying an appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm [HTF09].

In our case, the features are vectors containing color information in each attribute. This kind of data lies in a metric space, where two range values can be compared by the distance $d_j\left(x_{ji}, x_{ji'}\right) = \|x_{ji} - x_{ji'}\|$. Distance of two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ is the (possibly weighted) sum of their components. $D_{i,i'} = \sum_j^p w_j d_j \|x_{ji}, x_{ji'}\| = \|\mathbf{x}_i, \mathbf{x}_{i'}\|, \quad \sum_j^p w_j = 1$ . In our case we usually use either the Euclidean distance $\|\cdot\|_{\text{ssd}}$ or normalized cross correlation $\|\cdot\|_{\text{ncc}}$.

### 5.2.2. Clustering

We treat the vectors as data-points in a high-dimensional features space and use this representation for the analysis of the façade-image. Here we resort to mathematical methods of *unsupervised learning*, where the goal is to directly infer properties of the data-set and the underling probability density function (PDF) without any knowledge of correct data-samples or degree-of-error. In order to solve such problems the tool of choice is *cluster analysis*, which aims at grouping or segmenting a set of feature-vectors into subsets or clusters, such that features within one cluster are more closely related to each another than those assigned to other clusters [HTF09].

**K-Means.** One of the most known and widely used clustering algorithms is the K-means method. Its basic idea is straightforward: let the set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be all points in the data matrix $\mathbf{X}$. Given a random initial partitioning into $K$ clusters, compute the centroid of each cluster $\mathcal{C}_k$. The clusters are determined by minimizing the sum of squared errors:

$$E_k = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{m}_k\|_2^2 \quad \longrightarrow \min,$$

where $\mathbf{m}_k = \frac{1}{n_k} \sum_{i \in \mathcal{C}_k} \mathbf{x}_i$ is the centroid of the cluster $\mathcal{C}_k$ with $n_k$ points within. Then for each data-point $\mathbf{x}_i$ in a particular cluster, check whether there is another centroid that is closer than the present cluster centroid. If that is the case, then a redistribution is made. The algorithm usually has rather fast convergence, but one cannot guarantee that the algorithm finds the global minimum. The simplest heuristic to address this issue is to re-run K-means a chosen number of times, and to keep the solution with the best minimization result [HTF09].

**Spectral Clustering.** Spectral clustering is a generalization of standard clustering methods, and it is designed for situations where the data-points are not lying in convex clusters that can be grouped by spherical or elliptical metric. In spectral methods the actual clustering problem is casted as a graph partitioning problem, where we identify connected components with clusters. The data-points are represented as nodes and the similarity between the points represents the weights on the graph edges.

The simplest representation of such a graph is the adjacency matrix $\mathbf{A}$ that is a $n \times n$ symmetric matrix where each element $w_{ij}$ equals 1 if there is an edge between nodes $i$ and $j$ or 0 elsewhere. A more accurate representation is the affinity (also referred as similarity) matrix $\mathbf{W}$, where each element $w_{ij}$ stores some weight on the edge between nodes $i$ and $j$. Usually the weight is computed as a radial-kernel Gram* matrix, which is

$$w_{ij} = \exp\left(-d^2/\sigma^2\right)$$

with $\sigma > 0$ that is a scale parameter.

Now the graph can be partitioned, such that edges between different groups have low weight, and within a group have high weight. This structural relation is characterized by the spectrum of the graph, which can be obtained from the graph Laplacian matrix $\mathbf{L}$. To construct it, first one needs the diagonal degree matrix $\mathbf{D}$, whose each $(i,i)$-element is the sum of each row of $\mathbf{W}$. From the degree matrix and the similarity matrix one can construct the Laplacian matrix, where there are several possible approaches to define it

*Gram matrix is the matrix of all possible inner products of a set of vectors.

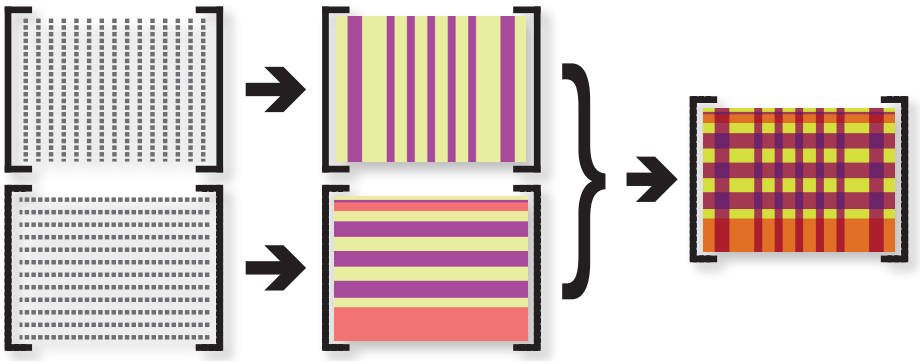

**Figure 5.3.:** *Clustering in both directions performed independently.*

(see [Lux07, WZ08] for an overview). For our purpose we have found that the normalized combinatorial Laplacian matrix as presented in [MS01] delivers best results:

$$\mathbf{L} = \mathbf{D}^{-1}\mathbf{W}.$$

The spectrum of the graph is obtained from the eigen-decomposition $\mathbf{L} = \mathbf{ZVZ}^{-1}$, where the diagonal of $\mathbf{V}$ contains the eigenvalues and the columns of $\mathbf{Z}$ are the eigenvectors of $\mathbf{L}$ respective. Note that $\mathbf{Z}$ is orthogonal. Depending on the number $K$ of expected clusters one needs to compute only $K$ eigenvectors corresponding to the $K$-largest (or smallest depending of the definition of $\mathbf{L}$ [NJW01]) eigenvectors. Now the eigenvectors form a matrix $\mathbf{U}$ of the size $m \times K$ and its rows can be interpreted as $m$ points $\in \mathbb{R}^K$ space, which can be now separated by ordinary K-means method. Finally we can assign each cluster label of each of those points to the original points in the matrix $\mathbf{X}$. Obviously, since we are working with very few clusters (usually 2-10) spectral clustering turns out to be, besides its other properties, an effective tool for dimensionality reduction.

### 5.2.3. Segmentation Optimization

Clustering the façade horizontal or vertical image line vectors into $K$ clusters and then labeling each vector according to its corresponding cluster does not guarantee that the labeling turn out to be spatially coherent, as illustrated in Figure 5.2 (c). Here we denote the image space along the horizontal and vertical axes as the spatial domain – in other words the order of the pixels in the image. This is due to the fact that neither the used metric nor the clustering algorithm itself take the spatial distribution of pixel-rows (or columns) into account.

Thus it is not ensured that similar pixel-rows in the sequence are grouped together into one cluster. In order to consider this issue, we reformulate the problem as a 1d Markov Random Field (MRF) and optimize the result of the clustering stage iteratively, such that pixel-rows

which are single (potential outlier) in the spatial domain are assigned to the nearest cluster with respect to both feature-space and spatial metric. In other words, we apply a "local force" on the pixel rows (columns) which holds coherent spatial gropes together and a "global force" that keeps account of proper clustering. Both forces are balanced such that we obtain an approximate optimal global solution for this problem.

Suppose we have a labeling problem where the task is to assign to each image row (or column) $\mathbf{x}_i$ some label from a finite label set $\mathcal{L}$ containing $K$ labels. Let $\mathcal{X}$ be the set of all rows (columns) in the façade image and $f(\mathbf{x})$ the label assigned to the row $\mathbf{x}$. Further, let $f = \{f_1(\mathbf{x}_i), \ldots, f_K(\mathbf{x}_i)\}_{i=1}^{n} \in \mathcal{F}$ be the collection of all possible labelings. Now we solve for the optimal labeling using the $\alpha$-expansion algorithm [BVZ01] by minimizing an energy functional $E$ over all possible horizontal (or vertical) image line vector labelings [KZ04, BK04];

$$E(f) = E_{\text{data}}(f) + \lambda \cdot E_{\text{smooth}}(f) \quad \longrightarrow \min,$$

where $\lambda$ expresses the relative confidence in the two terms of $E$. We define the data term $E_{\text{data}}(f)$ of $E$ to express the cumulative distance of each image line vector $\mathbf{x}_i$ from its assigned cluster centroid $\mathbf{m}_{f(\mathbf{x}_i)}$,

$$E_{\text{data}}(f) = \sum_{\mathbf{x}_i \in \mathcal{X}} \|\mathbf{x}_i - \mathbf{m}_{f(\mathbf{x}_i)}\|,$$

where $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\|\cdot\|$ is either $\|\cdot\|_2$ or $\|\cdot\|_{ncc}$. We define the smoothness term $E_{\text{smooth}}(f)$ of $E$ to promote spatial coherence for the cluster labels of adjacent image line vectors,

$$E_{\text{smooth}}(f) = \sum_{\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{N}} \begin{cases} 1 & \text{if } f(\mathbf{x}_i) \neq f(\mathbf{x}_j) \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{N}$ is the set of all adjacent pairs of image line vectors $\{\mathbf{x}_i, \mathbf{x}_j\} \subset \mathcal{X}$. Applying this optimization after clustering allows to correct iteratively the initial guess under the consideration of spatial coherence. Figure 5.2 (e) depicts the result of this regularization technique.

We apply the segmentation algorithm independently on both horizontal and vertical directions and combine the results into a grid as shown in Figure 5.3

## 5.3. Results

In this section we present several results of the clustering driven subdivision process. The value of $K$ mirrors the number of "different" types of elements which appear in one direction of the façade. We usually choose the number of cluster $K$ to be 2, which splits the image in "window" and "non-window" regions. In fact it depends on the façades' structure. In Figure 5.5 we show three examples of typical façades decomposed into a grid with

the value of $K = 2$ in the horizontal direction. In the vertical direction, we usually use the value of $K = 3$ that decomposes the façade in "window", "non-window" and "ground-floor" elements.

In Figure 5.4 we show an example of hierarchical splitting of the façade. First it has been decomposed in three regions: roof, floors and ground-floor. Than a separate tiling has been applied to each region. Currently we are providing the information of the number of clusters manually.

## 5.4. Conclusions

We presented an novel algorithm for robust façade segmentation based on unsupervised learning methodology. It performs on orthogonal and rectified façade imagery and provides quite stable segmentations without the usage of the notoriously error prone local edge detection. Our method involves global information and thus provides best segmentations with respect to the whole façade image. Its current limitation is the problem of the choice of the number of clusters which has to be selected by the user.

In future we plan to extend this approach to a inverse procedural modeling tool (cf. Section 2.6) such that the system will automatically obtain an appropriate value for the number of elements. We believe that rule sets, similar as proposed, e.g, by Aliaga *et al.* [ARB07], can be defined for a wide set of typical façades. We also want to encode the hierarchy which can be imposed over the façade model in a set of procedural rules. These will be than automatically chosen accordingly to the information obtained by the proposed clustering algorithm. Also its parameters will be derived in order to best fit the given façade image.

We believe that such automatic systems, combined with minor higher-level knowledge from the user, such as the number of potential clusters, will provide further contributions to the field of inverse procedural modeling.

**Figure 5.4.:** *Example of hierarchical clustering, where the roof, middle and ground-floors have been subdivided with an own set of parameters.*

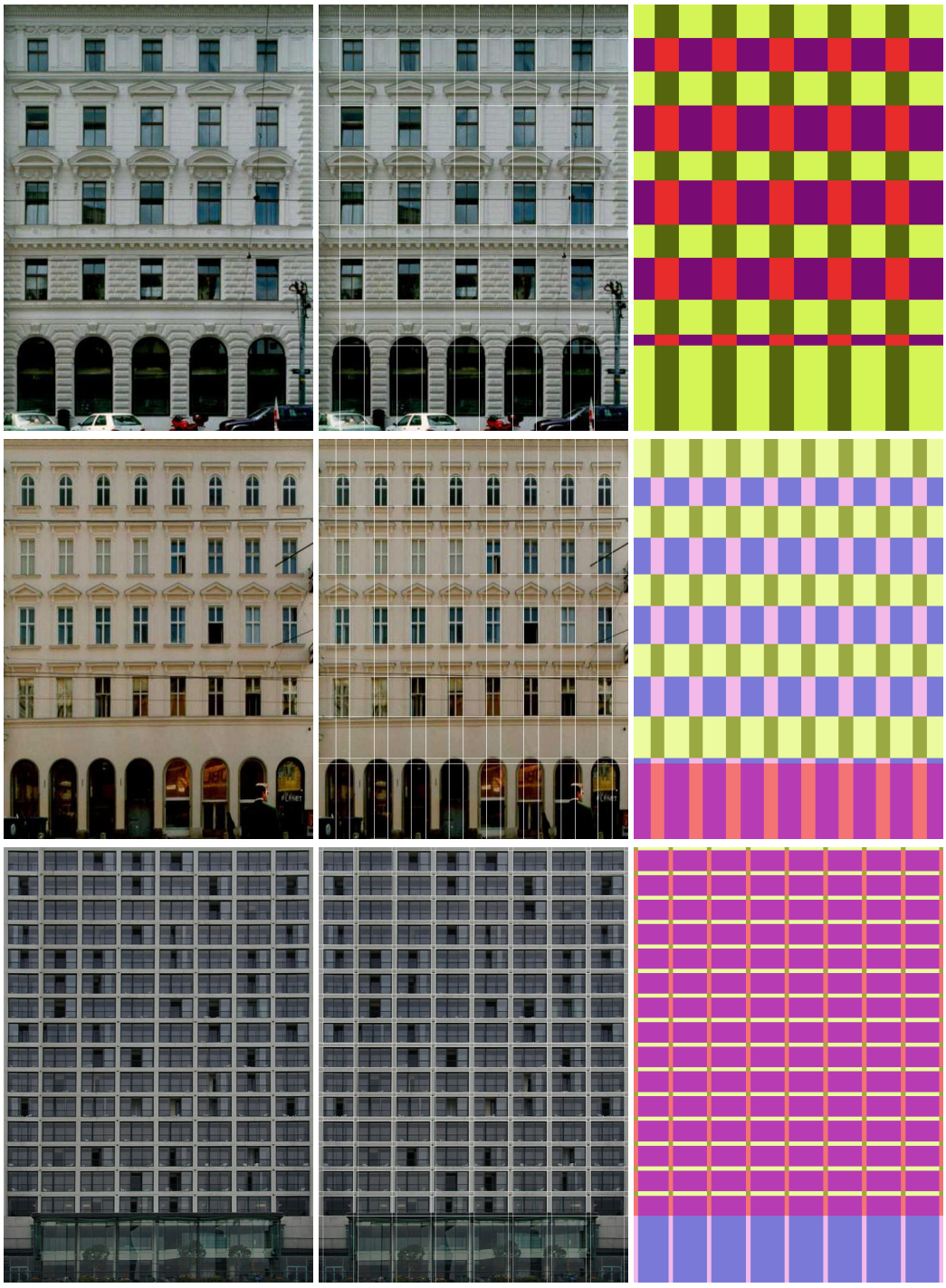

**Figure 5.5.:** *Examples of the segmentation algorithm. Clustering in both directions performed independently followed by Graph-Cut regularization. In all three cases the number of vertical clusters has been set to $K = 2$. In the upper example also the horizontal $K = 2$, the second and third example has the horizontal $K = 3$.*


**Figure 5.6.:** *Further results of the segmentation algorithm performed on various façade images. We used different combinations for the values of K but restricted it always only to* 2 *or* 3 *clusters.*

# 6. Façade Image Enhancement

In this chapter we address the problem of removing unwanted image content in a single orthographic façade image. We exploit the regular structure present in building façades and introduce a diffusion process that is guided by the symmetry prevalent in the image. It removes larger unwanted image objects such as traffic lights, street signs, or cables as well as smaller noise, such as reflections in the windows. The output is intended as source for textures in urban reconstruction projects.

## 6.1. Introduction

This chapter introduces a special image-processing method based on *symmetry propagation*. The proposed algorithm takes a single ortho-rectified façade image as input and tries to remove unwanted content, such as wall impurities, cables, and street signs (Figure 6.1). While this approach is similar to image in-painting in some respects, our goal and methodology are fairly different. First, we do not want to manually mark the irregularities by hand before removing them, but we would like to identify them automatically. Second, the focus of our algorithm is not a smooth transition or texture propagation from nearby regions, but structure propagation from detected symmetries. In principle, our algorithm is general and removes irregularities over a regular structure. While there are several potential applications for such an approach, the main motivation and probably most important use is the processing of façade images. Façade images are a vital component of three-dimensional urban reconstruction and we see applications in areas such as Internet and car-based mapping technology, urban simulation, and computer games.

Our contribution is twofold. First, we detect regular structures using a combination of Monte Carlo sampling and user interaction. Second, we use a diffusion-like process that tries to smooth across symmetries. It is a novel image processing technique that utilizes spatial symmetry in order to minimize asymmetric variations inherit in the image.

## 6.2. Overview

The basic idea behind this work is to exploit the repetitive occurrence of façade elements to reconstruct its clean and most plausible appearance. In order to handle the repetitive

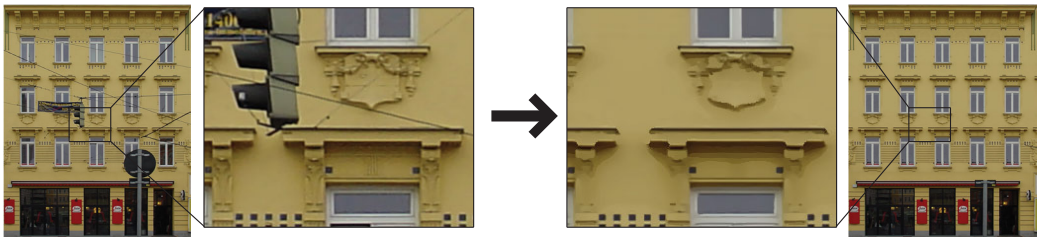

**Figure 6.1.:** *The input image on the left contains a traffic light and several cables. To the right we show the result of our algorithm with the unwanted objects successfully removed.*

nature we introduce the concept of a global symmetry neighborhood, which provides some basic information about the actual image. While there is conceptually no limitation on the topological complexity of the neighborhood-graph, we currently assume that the supplied image has strong translational and reflective repetitive elements as in the case of a building.

In section 6.3 we introduce a basic method to automatically determine the dominant symmetry in an image. The method is based on a Monte Carlo importance sampling strategy of image patch pairs and histogram evaluation and it yields the translational and reflective-translational symmetry in the rectified image.

In section 6.4 we introduce a novel method that utilizes the inherent symmetry in order to reconstruct the façade image. Missing or occluded elements, clutter and damage as well as small perspective distortion are dislodged and replaced by the information that can be accessed over the symmetry neighborhood in the entire image.

In section 6.5 we present and discuss restored façade images and finally we conclude our work in section 6.6.

## 6.3. Symmetry Detection

In our context, we define symmetry as a transformation $\mathbf{T}$ on an image. Given a pixel location $x$ of the input image $I$, we define $\mathbf{T}$ such that

$$I(x) = I(\mathbf{T}(x)) \tag{6.1}$$

where $I(x)$ denotes the intensity or color vector at $x$. As $\mathbf{T}$ we consider the 2-dimensional translations and reflections along the x-axis.

The goal of this stage is to determine the parameters of dominant transformations in the image automatically. For this purpose we refer to the algorithm presented in Chapter 4, which is a histogram voting scheme. The peaks of the histograms identify translational and reflective offsets in the image which occur most often and can be considered dominant. We use the dominant repetitions to define the transformations $\mathbf{T}$ for each pixel in the image. This allows us to define a symmetry neighborhood in form of a regular grid as depicted

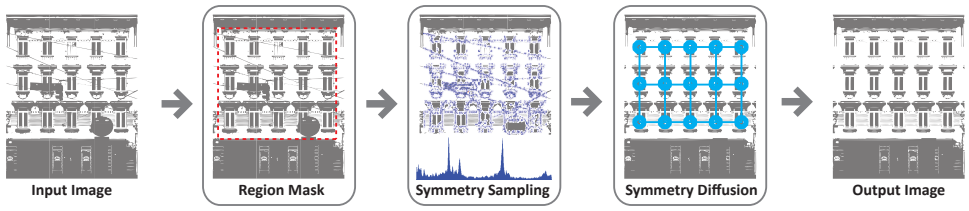


**Figure 6.2.:** *This figure shows an overview of the system. We take a single façade image as input I. For the case that the image does contain strongly asymmetric parts, we allow the user to define a region of interest. In this region we detect the dominant translational and reflective symmetries and propagate the symmetry over the image automatically.*

in Figure 6.3. Depending on the actual application, we can use all symmetry neighbor, or only those which lie either in horizontal or vertical or both direction to the current position. In praxis it has turned out that only these neighbors result in adequate coverage of the symmetry neighborhood.

## 6.4. Symmetry Propagation

### 6.4.1. Motivation

The symmetry propagation stage is the actual heart of our algorithm. Our idea behind this approach is motivated by the classical non-linear diffusion filter as presented by Perona and Malik [PM90]. They presented a powerful method for discontinuity-preserving smoothing and denoising of images based on a divergence equation

$$\frac{\partial I}{\partial t} = \operatorname{div}\left(g\left(\|\nabla I\|\right)\nabla I\right) , \tag{6.2}$$

where $g(x)$ is a flux-stopping function, which constrains the diffusion to pixels which have respectively small difference in range. It is usually of the form

$$g(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

where $x$ denotes the distance in the range. Later it has been shown [Bar01, Bar02] that this solution is also equivalent to the bilateral smoothing filter [TM98]. Hereby the basic idea is to apply a constrained Gaussian filtering to the image, such that steep range transitions become preserved. As a constraint an edge-penalty function has been introduced, which acts in principle the same way as the flux-stopping term mentioned above. The bilateral filter in a local neighborhood $\mathcal{N}_x$ of a pixel $x$ in image $I$ can be stated as:

$$I'_x = \frac{1}{W_x}\sum_{y\in\mathcal{N}_x} g_s\left(\|x-y\|\right) g_r\left(|I_x - I_y|\right) I_x , \tag{6.3}$$

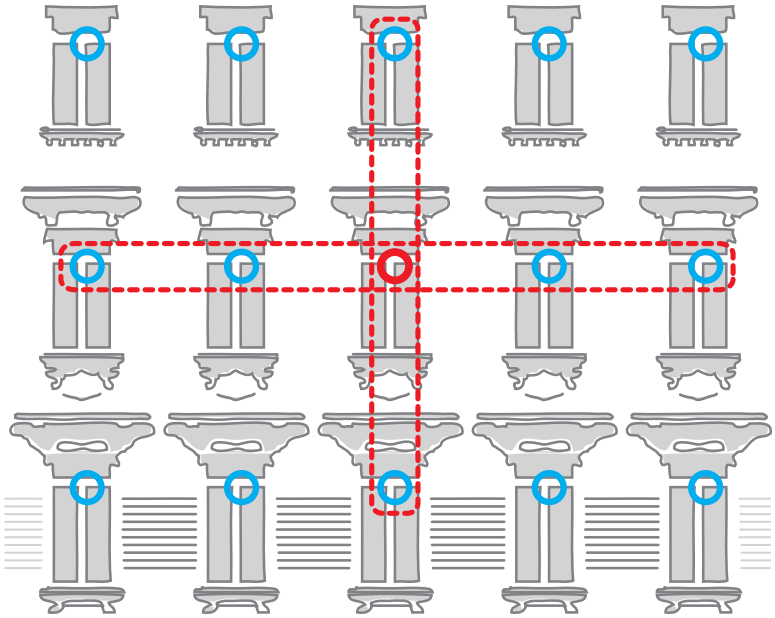

**Figure 6.3.:** *Symmetry Neighborhood. Blue circles denote symmetry neighbors of the red one.*

where

$$W_x = \sum_y g_s\left(\|x-y\|\right) g_r\left(|I_x - I_y|\right)$$

is a normalization term. There are two Gaussian functions in this equation: the usual $g_s$ acting in the spatial domain and $g_r$ applied on the range between the actually pixels values $I_x$ and $I_y$. The subscripts $s$ and $r$ denote the standard deviations $\sigma_s$ and $\sigma_r$ of the respective Gaussians. The result obtained is an image smoothed only in regions where the range difference is small enough to be emphasized by $G_r$.

More recently, non-local means filtering has been proposed as a new class of solutions to the image denoising problem [BCM05, BCM07, DFKE07]. It is based on the observation that pixels with similar neighborhood usually appear quite often in an image. Non-local filtering exploits this observation by computing a noise-reduced image by weighted averaging of many similar pixels:

$$I'_x = \frac{1}{W_x} \sum_{y \in I} w\left(x,y\right) I_x \,. \tag{6.4}$$

The term $W_x = \sum_{y \in I} w\left(x,y\right)$ is the normalizing constant, such that all $w(x,y) \in [0,1]$ and $\sum_y w(i,j) = 1$. The weights $w$ are computed according to the squared Euclidean norm of

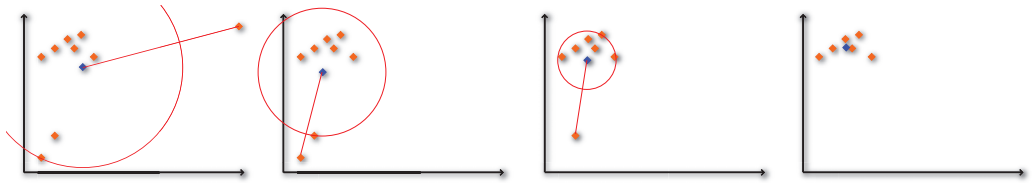

**Figure 6.4.:** *Recursive rejection of outliers. From left to right, at each step the point with the biggest distance to the mean (blue point) is removed and a new mean is computed, until the change is smaller than a given threshold.*

local neighborhoods $\mathcal{N}_x$ and $\mathcal{N}_y$, respectively. Writing the respective neighborhoods as vectors $\mathbf{x}$ and $\mathbf{y}$, the distance penalty function once again has the form:

$$w(x,y) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{h^2}\right).$$

The parameter $h$ acts as a degree of filtering [BCM07].

### 6.4.2. Iterative Symmetry Propagation

The approaches mentioned act (either locally or globally) on pixel neighborhoods in order to approximate a new image that is nearly noise-free, while a Gaussian penalty function is a common ingredient of these methods.

Our symmetry propagation algorithm is inspired by both bilateral- as well as non-local filtering. The main difference is that methods mentioned above aim at image repair by removing of noise that is a consequence of deficiencies in signal processing. We call it intrinsic noise. In contrast, our aim is the removal of both the intrinsic image noise as well as the extrinsic noise, e.g., traffic lights, cables, vegetation, missing elements and other interferences that are inherent in real world data. It is evident that the second class of noise can be removed only under certain circumstances: (1) there must be enough repeated content in the image and (2) there must be a strategy how to localize that information. The first one is a general assumption that for each location to be repaired there is enough information in the image which can be reused. For the second we resort to the symmetry and expect that the feasible information is arranged in a manner which can be expressed in terms of symmetry transformations $\mathbf{T}$ of the form $I(x) = I(\mathbf{T}(x))$.

In section 6.3 we have presented an elementary symmetry detection scheme. Having determined the global symmetry, each pixel $x$ in the image corresponds to a number of other pixels which can be addressed by the symmetry transformation $\mathbf{T}$ (see Equation 6.1). We shall refer to those pixels as the symmetry neighborhood $\mathcal{S}_x$ of the image location $x$. By the application of $\mathbf{T}$, it is possible to collect other neighbors and to obtain a set of points, which all correspond to a similarity in the image. Depending on the collection scheme, either all or only a subset of all possible symmetry neighbors can be accessed (see Fig. 6.3).

Having this information at a pixel, we now compute its actual consistency with its symmetry neighbors. Here we use different local neighborhood $\mathcal{N}_x$ as in the symmetry sampling stage from section 6.3. We have determined empirically that sizes between $3 \times 3$ and $11 \times 11$ deliver reasonable results for our input images. With $\mathcal{N}_x$ as a vector $\mathbf{x}$ of intensity values we can compute the mean vector for all points $\mathbf{x}_i \in \mathcal{S}_x$ with $n = |\mathcal{S}_x|$ as

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i}^{n} \mathbf{x}_i \,.$$

If we would like to apply unconstrained symmetry propagation to the image, the actual new color value for the output pixel $I'_x$ would be the middle element in $\bar{\mathbf{x}}$, which also equals the simple average color over all symmetry neighbors. But we are interested in some constraints, which will allow us to determine which of the pixels belong to the most symmetric façade image and which are potentially clutter. Following the first assumption that the majority of the pixels in $\mathcal{S}_x$ contain valid values, we introduce a scheme inspired by Expectation Maximization to reject outliers. It utilizes the fact that if there are more valid pixels they will also be more similar and thus lie denser to each other in the space defined by the local neighborhood vectors $\mathbf{x}_i$. In this case we define the outlier as the point which has the biggest distance to the mean $\bar{\mathbf{x}}$:

$$\mathbf{x}_i = \underset{\mathbf{x}_i \in \mathcal{S}_x}{\arg\max} K\left(\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2\right) \,, \tag{6.5}$$

where $K$ is a Gaussian kernel. Now we can remove the $i$-th vector from $\mathcal{S}_x$ and recompute the mean $\bar{\mathbf{x}}$. We define the difference of the mean as:

$$\mathbf{m}_{i+} = \frac{\sum_i \mathbf{x}_i K\left(\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2\right)}{\sum_i K\left(\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2\right)} - \bar{\mathbf{x}} \,, \tag{6.6}$$

which is basically the mean-shift [CM02]. We use this vector to determine maximal mode of the distribution of the points in the $\mathcal{S}_x$. We proceed iteratively until either the mean $\bar{\mathbf{x}}$ does not change more than a given threshold $\varepsilon$ or only one point is left in $\mathcal{S}_x$. We measure the change as $|\mathbf{m}|$, which gives a indication how much does the pixel alters after each iteration. Using a sufficiently small $\varepsilon$ this procedure delivers the most dense cluster of the symmetry neighborhood (see Fig. 6.4).

To determine the final color of the output pixel we additionally apply bilateral filtering over the local neighborhoods remained in $\mathcal{S}_x$ after the optimization, such that the output pixel value is:

$$I'_x = \frac{1}{W_x} \sum_{\mathcal{S}} \sum_{\mathcal{N}} g_s\left(\|x - y\|\right) g_r\left(|I_x - I_y|\right) I_x \,, \tag{6.7}$$

where $W_x$ is an appropriate weight according to equation 6.3. We do this in order to smooth possibly remaining variations caused by inaccuracy of the symmetry transformation.

Image parts, which violate the detected symmetry are replaced by pixels which become amplified by strong symmetry. In case of strong asymmetries some of them can still remain in the image after the first iteration. In this case we apply further passes of the algorithm until no more changes can be observed in the image. This is usually already the case after the second iteration, as shown in Figure 6.5.

## 6.5. Results

We have implemented the algorithm in a mixture of C# and MATLAB and ran it on an Intel Core2 Quad Q6600 @ 2.4 GHz, 8 GB RAM and Vista64 computer. We show image pairs of input façades and the result of our symmetry propagation in Figures 6.6, 6.7 and 6.8. The running times for the first four examples are reported in Table 6.1. Note that the running time depends not only on the size of the image, but also on the number of symmetry neighbors and the degree of distortion. On images with a large symmetry neighborhood the running time takes up to several minuets. As an example observe image #3, whose running time is shorter than, e.g., image #2 in spite of the former's lower resolution. This is due to the quite large symmetry neighborhood of the façade and the variable number of iterations of the outlier rejection routine.

The last example in Figure 6.9, depicts a failure case of our algorithm. In the upper story the outlier rejection method could not determine the actual wall color coherently.

## 6.6. Discussion and Conclusions

### 6.6.1. Limitations

The optimization technique presented in section 6.4 does not always converge to an optimum. While this is usually not a big problem, the bad situation occurs when the symmetry neighborhood of a pixel contains two or more (roughly) equally balanced clusters. In this case it is not guaranteed that the algorithm converges to the right configuration. Furthermore, neighboring pixels might converge at different clusters, which yields strong artifacts in the image, as shown in Figure 6.9 (septically the upper story of the building). We are

| Image | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| resolution | 1140x1420 | 1802x1160 | 990x1400 | 548x884 |
| $3 \times 3$ | 4.3 | 8.5 | 12.2 | 3.1 |
| $11 \times 11$ | 41.7 | 61.2 | 87.8 | 34.1 |

**Table 6.1.:** *Comparison of the running times (given in seconds) for the first 4 images presented in Figures 6.6 and 6.7. The images were computed with a local neighborhood of* $3 \times 3$ *and* $11 \times 11$ *pixels over 5 iterations.*

currently working on an improved optimization method for our algorithm as well as on a more flexible symmetry detection strategy.

### 6.6.2. Conclusions

In this chapter we presented a method to remove irregularities in a single approximately orthographic façade image using a symmetry propagation process. The symmetry is first detected using Monte Carlo sampling and encoded in a symmetry neighborhood. The symmetry is then propagated while performing edge-preserving smoothing on the image. This method can remove unwanted features, such as traffic lights, cables, signs and cars that are typically present in a façade image. It is not necessary to manually segment the unwanted image elements prior to running the algorithm, except for providing a coarse region mask. The output is intended to serve as input to rendering pipelines such as, e.g., Ali *et al.* [AYRW09]. We believe that our work is a useful solution to an important image processing step necessary in urban reconstruction projects.

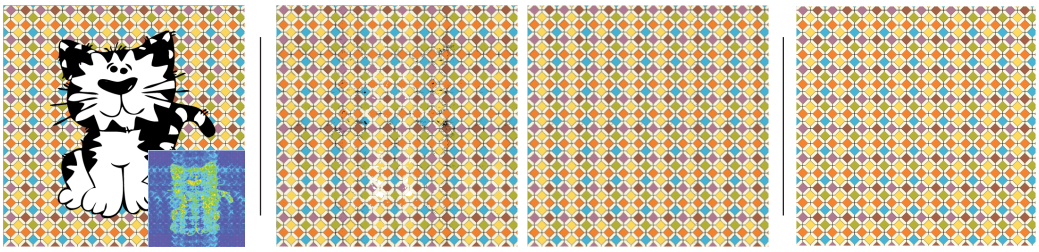

**Figure 6.5.:** *Left: An artificial test image with high symmetry (with a symmetry confidence map). Middle: The distortion has been removed in two iterations. Right: The ground truth.*

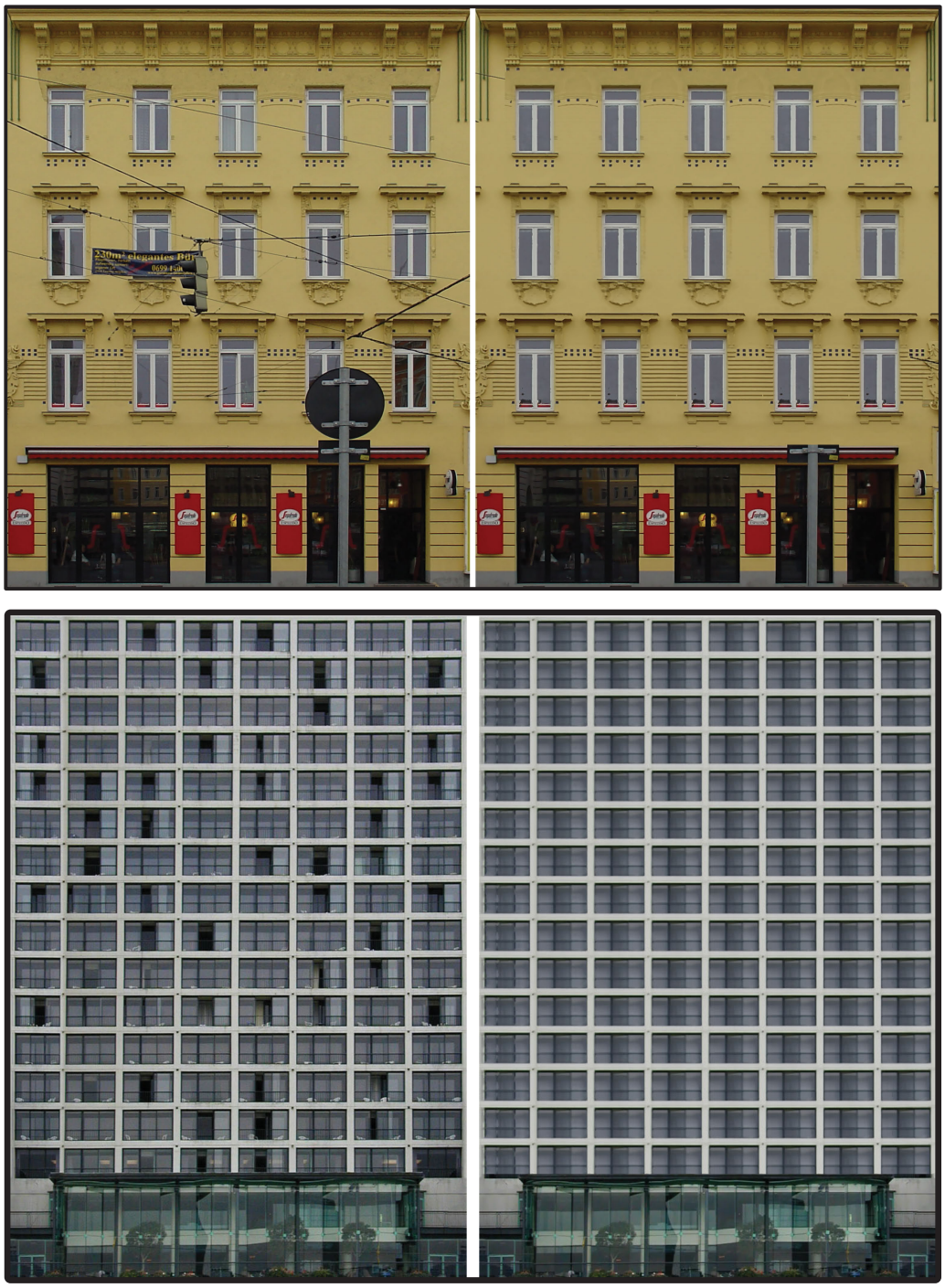

**Figure 6.6.:** *We show image pairs of input façades and the result of our symmetry propagation.*

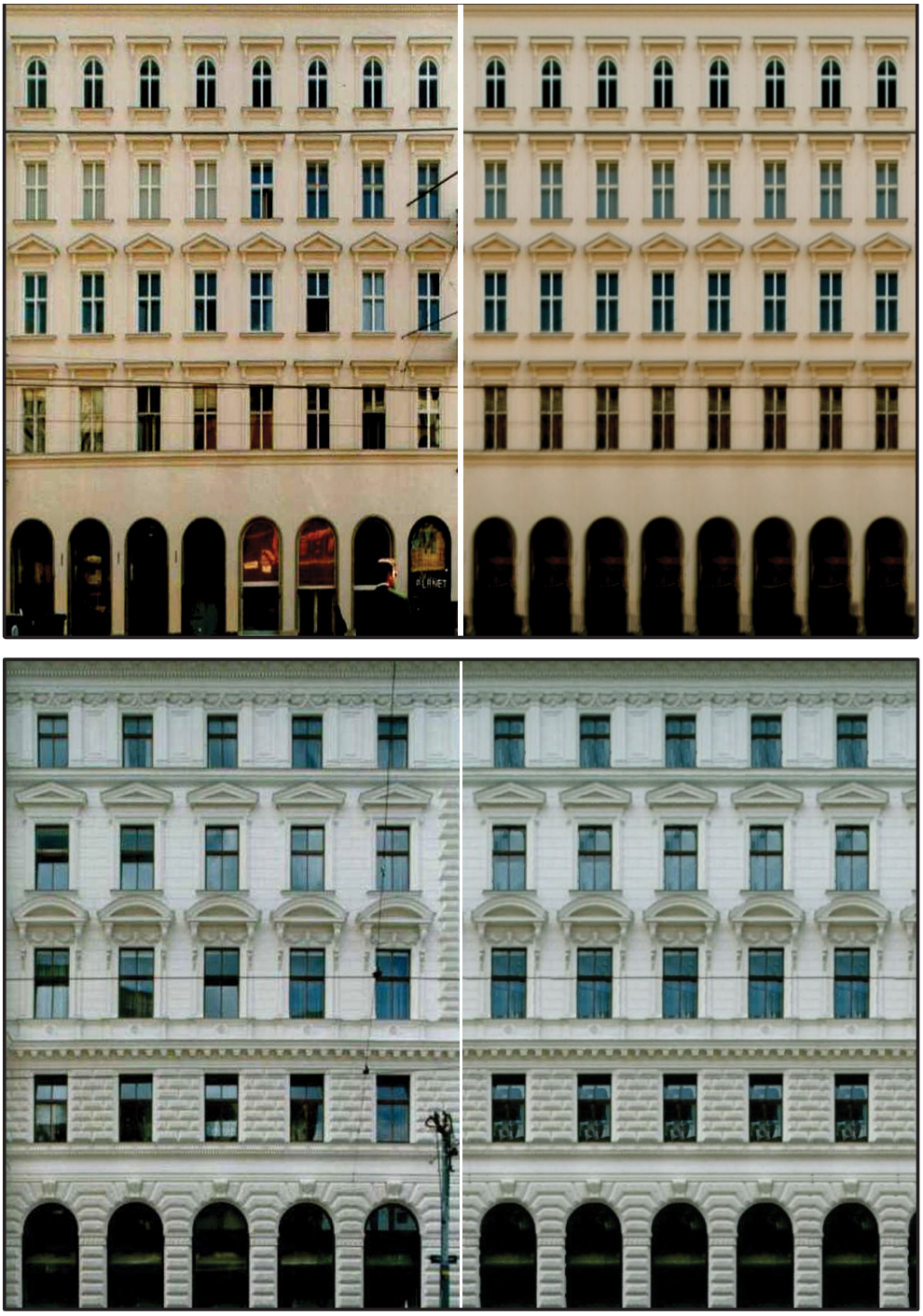

**Figure 6.7.:** *We show image pairs of input façades and the result of our symmetry propagation.*


**Figure 6.8.:** *We show image pairs of input façades and the result of our symmetry propagation.*


**Figure 6.9.:** *A failure case of our algorithm. In the upper story the outlier rejection method could not determine the actual wall color coherently, which results in destroyed structure of the original image..*

# 7. Conclusions and Outlook

In this dissertation we propose a set of algorithms for processing of façade imagery in the context of urban reconstruction. The aim of the algorithms is to provide improvements to the current state-of-the-art in the generation of high-quality façade textures. In particular, we show that we can generate high-quality images which are not possible to be taken by conventional image acquisition techniques.

## 7.1. Conclusions

In Chapter 3 we present a complete pipeline for generating of orthogonal façade images. We show the details of the approach and propose a solution, where we combine image processing and reconstruction algorithms, such as structure-from-motion and poisson image blending, and build novel approaches on top of this methodology in order to remove obstacles in the façade image. One of them is automatic based on a heuristic, the two others are user assisted. It turns out that even if the automatic approach often provides very good results, in order to achieve the best quality, user interaction is a necessary tool of the pipeline. In this context it has to be mentioned that user interaction in modern computer assisted vision applications can be implemented in a very efficient and subtle manner. The goal should be to provide the user simple interaction tools, like brushes which can be used even by an unskilled user.

Another key contribution of this thesis are two approaches to façade segmentation. The method presented in Chapter 4 is based on random sampling of image features in order to detect dominant repetitions. It is inspired by the family of symmetry sampling algorithms based on voting schemes. Our method introduces simplifications which take the specific properties of the rectangular nature of façades into account.

The novel approach for façade segmentation presented in Chapter 5 introduces a method not documented in the literature before. It is inspired by matrix factorization, data mining and generally by the linear algebra of the field of unsupervised learning [HTF09]. Our method provides a simple yet impressive way to segment architectural (thus mainly rectangular and axis-aligned) imagery very efficiently. We believe that this approach will serve as a basis for further urban modeling solutions.

In general, the problem of façade segmentation is much harder than expected. The main reason is that it is not clear if there exists a general top-down model for all façades. Moreover, this task naturally suffers under the "chicken or the egg" dilemma and it is in general an unsolvable task. Segmentation algorithms depend on the context, and this is an important conclusion of this work. We have taken this issue into account and propose methods that exploit special higher level knowledge in order to solve the problem and, if necessary, minimal user interaction. We believe that this strategy is the optimal one if the goal is the generation of high-quality segmentations with a minimal number of errors.

In Chapter 6 we present a novel image processing method which we call symmetry propagation. This idea is inspired by both the rectangular and the repetitive nature of façades. We develop the theory behind the method and show its effects on a number of orthogonal façade images. Our basic idea is quite general and the removal of occlusions by the means of symmetry information is a vital idea that inspired others in order to process laser scan data [ZSW*10]. Architectural imagery is ideally suited for such approaches, but also other, more general image processing algorithms that exploit this clue are still under development [CZM*10].

Additionally, the presented dissertation provides a comprehensive overview over the wide spread and rather young research field of image-based modeling and reconstruction. It intentionally balances on the border between graphics and vision in order to grasp the best of both worlds – the quality of graphics and the automation of vision. Even if this is not fully possible, we believe that this research direction is target-aimed and leads to successful development of next generation interactive image-based modeling tools for the virtual reconstruction of our world.

## 7.2. Outlook

This thesis is the result of over three years of research on the topic of image based-urban reconstruction. The presented contributions as well as the extensive studies of a huge collection of related literature allows us to provide an outlook into the future of the addressed research field.

Urban reconstruction is by far not solved yet. For the future there is still a lot of work to be done. Even if there has been significant progress in the field of single-image and recently also in multi-image processing, such robust methods actually open the door for further, advanced image processing algorithms.

Segmentation by matrix factorization is an unexplored topic. The approach in this thesis gives a seeding building block for this topic, but we believe that higher-order knowledge of architecture in combination with image segmentation in context of unsupervised, robust methods is a promising combination and will provide solutions in the future.

However, a probably even more undeveloped topic is inverse procedural modeling. For urban reconstruction grammar driven approaches promise semantic segmentation, but the

approach is still not well defined and an automatic solution is in its infancy. Nevertheless, we believe that this approach has a future in reasonable time.

One essential problem is the integration of the research on the reconstruction of the world. Besides the concurring global commercial companies, there is also a slight divergence in the scientific fields. We mean here the parallel research in the computer science disciplines (CG and CV) and photogrammetry and remote sensing field. In particular, problems of storage, GIS, GPS and geo-registration as well as the cooperation of researchers of computer sciences and photogrammetry and remote sensing could be improved. This thesis tries to contribute to this idea by providing an interdisciplinary literature review composed of works of all of the mentioned fields.

# A. Homography

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.

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

**Calculating the Homography from Corresponding Image Points** Using the homography we can project the perspective image into a corresponding orthogonalized image: $\mathbf{p}'_b = \mathbf{H}_{ab}\mathbf{p}_a$. 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 $\mathbf{P}_i$:

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

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

$$x_i\mathbf{P}_3\mathbf{a}_i - z_i\mathbf{P}_1\mathbf{a}_i = 0$$

$$y_i\mathbf{P}_3\mathbf{a}_i - z_i\mathbf{P}_2\mathbf{a}_i = 0$$

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

# B. Point Cloud to Model Registration

It is easy to register the point-cloud in the model manually until a certain degree of accuracy - one can move, scale and rotate the point-cloud in a CAD or modeling software very easy into the near of its proper extends. Unfortunately it is an almost impossible task to do it accurate enough for any useful application. Here we present registrations algorithms, which allow us to solve such a task under certain conditions.

This report will present and compare three of them: The classical Iterated Closed Points (also referred as Iterated Corresponding Points) algorithm [BM92] based on the rigid 3d registration method of Horn [Hor87]. The second and the third are based on the vector-field of the motion, which has to be traversed by the the point-cloud in order to move from its initial position into the optimal one. This methods will be elaborated in more detail in section B.

## The Horn-Method (ICP)

The actual Horn method [Hor87] was intended to compute the similarity transformation between two point-clouds in different coordinate systems. Basically, one needs only three non-coplanar corresponding points of both systems to do so. Of course one can compute a least-square solution of a larger set of corresponding points. Than, the similarity transformation is split into the three basic operations: translation and rotation (for the Euclidean transformation) and scale (for a similarity). Horn proposes to compute the translation vector $\mathbf{t}$ of two corresponding point-clouds $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ by the means of their (possibly weighted) barycenters:

$$\mathbf{t} = \mathbf{c}_y - \mathbf{c}_x = \frac{1}{W}\sum_i w_i \mathbf{y}_i - \frac{1}{W}\sum_i w_i \mathbf{x}_i,$$

with $W = \sum_i w_i$. The vector $\mathbf{t}$ gives us the translational relationship between the point-sets. For further convenience, we can also translate both point-clouds into the origin. Then, we can determine the rotational part of the mapping explicitly as:

$$\mathbf{x}' = \mathbf{R}\mathbf{x}\,. \tag{B.1}$$

Horn proposes to rewrite the rotation into an unit quaternion. With this approach we can rewrite:

$$\sum_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i = \mathbf{a}^T \cdot \sum_i \left(\mathbf{Y}_i^T \cdot \tilde{\mathbf{X}}_i\right) \cdot \mathbf{a} = \mathbf{a}^T \mathbf{M}\mathbf{a} \quad \longrightarrow \max \ . \tag{B.2}$$

This maximizes the quadratic form under the quadratic constraint

$$\|a\|^2 = \mathbf{a}^T \mathbf{a} = 1 \ ,$$

which leads to the solution of a general eigenvalue problem. The largest eigenvalue of the matrix $\mathbf{M}$ of Equation B.2 gives the actual solution as a quaternion $\mathbf{a} \in \mathbb{R}^4$. By the application of the rotation of the quaternion to the point-cloud we have determined the optimal rigid transformation of both point-clouds. For further details as well as how to determine the scale in order to perform similarity mappings refer to Horn [Hor87].

**ICP** While the Horn-Method works basically only with given point-to-point correspondences, it can be easily extended to the iterated closest point approach. We do so by iteratively alternating of the two steps: translation and rotation computation. This method is one version of the ICP algorithm, and the objective function can be defined as:

$$F = \sum_i \|\mathbf{x}_{i+} - \mathbf{y}_i\|^2 \quad \longrightarrow \min \ . \tag{B.3}$$

Between each iteration we need to compute new point-to-point correspondences $\mathbf{x}_{i+}$ and $\mathbf{y}_i$. For big data sets this is usually the computationally most expensive task and can be countered with accelerations for nearest neighbor search, e.g., kd-trees.

## The Helical Motion Method

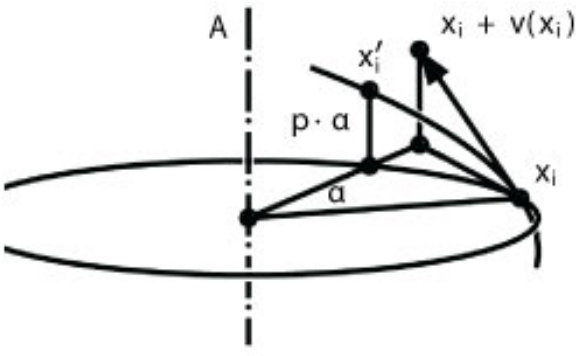


**Figure B.1.:** *Motion along a helix.*

**Point-to-Point.** The Helical-Motion Algorithm is inspired by the track which a rigid body has to traverse to reach the optimal position with respect to a reference point-cloud. During this journey the body moves iteratively along helices: it translates and rotates along some axis: the goal is to determine this axis and the angular velocity. This motion can by formalized by a vector field of the form:

$$\mathbf{v}(\mathbf{x}) = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}. \tag{B.4}$$

The unknowns are the both vectors $\mathbf{c}$ and $\bar{\mathbf{c}}$. By linearizing the problem we can approximate the new positions of the points by $\mathbf{x}' = \mathbf{x}_i + \mathbf{v}(\mathbf{x}_i)$. Now the objective function can be formulated as:

$$\sum_i (\mathbf{x}_i + \mathbf{v}(\mathbf{x}_i) - \mathbf{y}_i)^2 = \sum_i (\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i + \mathbf{x}_i - \mathbf{y}_i)^2 \ . \tag{B.5}$$

This quadratic function can be solved explicitly by a system of linear equations. To do so, we can formulate the problem in matrix form as:

$$\begin{aligned} F &= \sum_i (\mathbf{c} \times \mathbf{x}_i + \bar{\mathbf{c}} + \mathbf{x}_i - \mathbf{y}_i)^2 \\ &= \mathbf{C}^T \mathbf{A} \mathbf{C} + 2\mathbf{B}^T \mathbf{C} + \mathbf{D} \end{aligned} \tag{B.6}$$

where

$$\mathbf{B} = \sum_i \begin{bmatrix} \mathbf{x}_i \times (\mathbf{x}_i - \mathbf{y}_i) \\ \mathbf{x}_i - \mathbf{y}_i \end{bmatrix}$$

and $\mathbf{D}$ is a constant:

$$\mathbf{D} = \sum_i (\mathbf{x}_i - \mathbf{y}_i)^2 \ .$$

The more tricky part is the matrix $\mathbf{A}$ which is the normal equation of the matrix $\mathbf{M}^T\mathbf{M}$, where we can factorize the cross-product by the skew-symmetric matrix $[\mathbf{x}]_\times$:

$$\mathbf{M} = \begin{bmatrix} [\mathbf{x}]_\times & \mathbf{I} \end{bmatrix} = \begin{bmatrix} 0 & x_z & -x_y & 1 & 0 & 0 \\ -x_z & 0 & x_x & 0 & 1 & 0 \\ x_y & -x_x & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{B.7}$$

Finally, the unknowns vector is defined as a six-by-one matrix:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c} \\ \bar{\mathbf{c}} \end{bmatrix} \tag{B.8}$$

and we solve a system of six linear equations as:

$$\mathbf{C}^T \mathbf{M}^T \mathbf{M} \mathbf{C} = \mathbf{A}\mathbf{C} = -\mathbf{B}$$

which equals

$$\mathbf{C} = \mathbf{A}^{-1} (-\mathbf{B}) \ . \tag{B.9}$$

This system can be easily solved by i.e. LU-decomposition of the matrix $\mathbf{A}$.

**Point-to-Plane.** A second registration algorithm based on the idea of helical-motion of a rigid body is equivalent to the Gauss-Newton iterative solution. The main advantage of this approach is that it converges quadratically to the (local) minimum [PLH04]. The idea here is to minimize the squared distance between the points of the data point cloud and the tangent planes of the reference surface. To do so, we have to compute the closest point-pairs as in the algorithm before. Than we need to determine a tangent plane on each foot-point on the surface and drop a perpendicular from point $x_i$ onto this tangent plane. Now the goal is to minimize the square of this distance iteratively. Let us denote the two corresponding points as $\mathbf{x}_i$ and $\mathbf{y}_i$. The normal can be defined as

$$\mathbf{n}_i = \frac{\mathbf{x}_i - \mathbf{y}_i}{\|\mathbf{x}_i - \mathbf{y}_i\|} \,. \tag{B.10}$$

The location of the point $\mathbf{x}_i$ can be approximated by the velocity field $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{v}(\mathbf{x}_i)$ as defined in equation B.4. The squared distance to the tangent plane is given by:

$$d^2(\mathbf{x}'_i) = \left(\left(\mathbf{x}'_i - \mathbf{y}_i\right) \cdot \mathbf{n}_i\right)^2 = \left((\mathbf{c} \times \mathbf{x}_i + \bar{\mathbf{c}} + \mathbf{x}_i - \mathbf{y}_i) \cdot \mathbf{n}_i\right)^2 = \left(\bar{\mathbf{c}}\mathbf{n}_i + \mathbf{c}\bar{\mathbf{n}}_i + d_i\right)^2 \,, \tag{B.11}$$

where $\bar{\mathbf{n}}_i = \mathbf{x}_i \times \mathbf{n}_i$. The objective function is finally then:

$$\begin{aligned} F &= \sum_i \left(\bar{\mathbf{c}}\mathbf{n}_i + \mathbf{c}\bar{\mathbf{n}}_i + d_i\right)^2 \\ &= \mathbf{C}^T\mathbf{A}\mathbf{C} + 2\mathbf{B}^T\mathbf{C} + \mathbf{D} \,. \end{aligned} \tag{B.12}$$

The matrix $\mathbf{A}$ is the normal equation of the objective function, where $\mathbf{a}_i$ is a one-by-six vector $\mathbf{a}_i = [\mathbf{x}_i \times \mathbf{n}_i, \mathbf{n}_i]$. Thus the matrix $\mathbf{A}$ a six-by-six matrix:

$$\mathbf{A} = \sum_i \mathbf{a}_i^T \mathbf{a}_i$$

The column vector $\mathbf{B}$ is a six-by-one vector:

$$\mathbf{B} = \sum_i d_i \mathbf{a}_i^T \,.$$

Finally, we can solve this system linearly as in the algorithm before as $\mathbf{AC} + \mathbf{B} = \mathbf{0}$, where $\mathbf{C}$ is the same unknown vector as in equation B.8. For more details on this algorithm refer to [PLH04] and for its analysis to [PHYH06].

**Transformation.** The two algorithms above end up with the both vectors $\mathbf{c}$ and $\bar{\mathbf{c}}$. In order to determine the transformation matrix we can use the Plücker coordinates along the trajectory (for detail refer to [PLH04] and Fig. B.1). According to Figure B.1, the Plücker

coordinates $(\mathbf{g}, \bar{\mathbf{g}})$ of the axis $A$, the pitch $p$ and the angular velocity $\omega$ are computed from $(\mathbf{c}, \bar{\mathbf{c}})$ as:

$$\mathbf{g} = \frac{\mathbf{c}}{\|\mathbf{c}\|} \qquad \bar{\mathbf{g}} = \frac{\bar{\mathbf{c}} - p\mathbf{c}}{\|\mathbf{c}\|} \qquad p = \frac{\bar{\mathbf{c}}\mathbf{c}}{\mathbf{c}^2} \qquad \omega = \|\mathbf{c}\| \,.$$

## Summary

During the implementation and the experiments it has turned out that all three algorithms have their pros and cons. The Horn-Method performs basically quite well, especially for pure rotations. Its disadvantages come out in cases where one wants to register point-clouds containing non-uniform spaced points. Due to the barycenter-based translation it does not always converge properly.

The Helical-Motion algorithms are very interesting interpretations of the problem. Basically it turned out that the point-to-point, linearly converging algorithm has the most worst performance: is converges quite slowly, never as close to the reference data as the point-to-plane version and it also often gets trapped into a local minimum. Nevertheless, it is still a practical algorithm, especially for the case of iterative registration of more the two data sets [PLH04]. Finally, the Gauss-Newton equivalent point-to-tangent-plane version of the algorithm has surely best performance: it converges quite fast and also closer to the reference than any other of the algorithms. Of course it also can get stuck in an local minimum. This problem is acute to all of the algorithms and it should be overcome by providing proper starting conditions to all of them. This can be done manually, as in the case of the Stephansdom dataset as well it could be accomplished by stochastic algorithms as Monte-Carlo-Markov-Chain solved by Simulated Annealing (MCMC).

A short video to this appendix can be downloaded at:
**http://www.youtube.com/watch?v=MxyyZ0907nM**

# References


[AAC*06] AGARWALA A., AGRAWALA M., COHEN M., SALESIN D., SZELISKI R.: Photographing long scenes with multi-viewpoint panoramas. *ACM Transactions on Graphics 25*, 3 (July 2006), 853. 9

[ABVA08] ALIAGA D. G., BENEŠ B., VANEGAS C. A., ANDRYSCO N.: Interactive Reconfiguration of Urban Layouts. *IEEE Computer Graphics and Applications 28*, 3 (May 2008), 38–47. 21

[AD04] ALEGRE F., DELLAERT F.: A Probabilistic Approach to the Semantic Interpretation of Building Facades. In *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres, 2004* (2004). 25

[ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 294. 10, 38

[AFM*06] AKBARZADEH A., FRAHM J.-M., MORDOHAI P., CLIPP B., ENGELS C., GALLUP D., MERRELL P., PHELPS M., SINHA S. N., TALTON B., WANG L., YANG Q., STEWENIUS H., YANG R., WELCH G., TOWLES H., NISTER D., POLLEFEYS M.: Towards Urban 3D Reconstruction from Video. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)* (June 2006), IEEE, pp. 1–8. 28

[Aga07] AGARWALA A.: Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics 26*, 3 (July 2007), 94. 10

[AR07] AGRAWAL A., RASKAR R.: Gradient Domain Manipulation Techniques in Vision and Graphics. ICCV 2007 Course (\url{http://www.umiacs.umd.edu/~aagrawal/ICCV2007Course/index.html}), 2007. 10, 40

[ARB07] ALIAGA D. G., ROSEN P. A., BEKINS D. R.: Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 786–97. 20, 23, 24, 73

[arc10] Automatic Reconstruction Conduit. **http://www.arc3d.be/**, October 2010. 16

[ASJ*07] ALI H., SEIFERT C., JINDAL N., PALETTA L., PAAR G.: Window Detection in Facades. In *14th International Conference on Image Analysis and Processing (ICIAP 2007)* (Sept. 2007), IEEE, pp. 837–842. 14

[ASS*09] AGARWAL S., SNAVELY N., SIMON I., SEITZ S. M., SZELISKI R.: Building Rome in a day. In *2009 IEEE 12th International Conference on Computer Vision* (Sept. 2009), IEEE, pp. 72–79. 16

[AYRW09] ALI S., YE J., RAZDAN A., WONKA P.: Compressed facade displacement maps. *IEEE Transactions on Visualization and Computer Graphics 15*, 2 (2009), 262–73. 12, 84

[BA83] BURT P. J., ADELSON E. H.: A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics 2*, 4 (Oct. 1983), 217–236. 10

[Bai97] BAILEY D. G.: Detecting regular patterns using frequency domain self-filtering. IEEE Computer Society. 11

[Bar01] BARASH D.: Bilateral Filtering and Anisotropic Difusion: Towards a Unified Viewpoint. In *Scale-Space and Morphology in Computer Vision* (Berlin, Heidelberg, June 2001), vol. 2106/2001 of *Lecture Notes in Computer Science 2106*, Springer, pp. 273–280. 79

[Bar02] BARASH D.: Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 6 (June 2002), 844–847. 79

[BBM*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured lumigraph rendering. *International Conference on Computer Graphics and Interactive Techniques* (2001). 20

[BCM05] BUADES A., COLL B., MOREL J.-M.: A Non-Local Algorithm for Image Denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 2, IEEE, pp. 60–65. 80

[BCM07] BUADES A., COLL B., MOREL J.-M.: Nonlocal Image and Movie Denoising. *International Journal of Computer Vision 76*, 2 (July 2007), 123–139. 80, 81

[Bec09] BECKER S.: Generation and application of rules for quality dependent façade reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing 64*, 6 (Nov. 2009), 640–653. 26

[BETvG08] BAY H., ESS A., TUYTELAARS T., VAN GOOL L.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding 110*, 3 (June 2008), 346–359. 15

[BFM08a] BALDAUF M., FRÖHLICH P., MUSIALSKI P.: A Lightweight 3D Visualization Approach for Mobile City Exploration. In *First International Workshop on Trends in Pervasive and Ubiquitous Geotechnology and Geoinformation GIScience conference (TIPUGG'08)* (2008). 5

[BFM08b] BALDAUF M., FRÖHLICH P., MUSIALSKI P.: Integrating User-Generated Content and Pervasive Communications - WikiVienna: Community-Based City Reconstruction. *IEEE Pervasive Computing 7*, 4 (Oct. 2008), 58–61. 5

[BH07] BECKER S., HAALA N.: Refinement of Building Facades by Integrated Processing of LIDAR and Image Data. In *PIA07. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (Sept. 2007), University of Stuttgart : Collaborative Research Center SFB 627 (Nexus: World Models for Mobile Context-Based Systems). 26

[BH09] BECKER S., HAALA N.: Grammar supported facade reconstruction from mobile lidar mapping. In *ISPRS Workshop, CMRT09 - City Models, Roads and Traffic* (2009), vol. XXXVIII. 26

[BHF08] BECKER S., HAALA N., FRITSCH D.: Combined knowledge propagation for facade reconstruction. In *ISPRS Congress Beijing 2008, Proceedings of Commission V* (2008). 26

[BI07] BOIMAN O., IRANI M.: Detecting Irregularities in Images and in Video. *International Journal of Computer Vision 74*, 1 (Jan. 2007), 17–31. 11

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 9 (Sept. 2004), 1124–37. 72

[BKS*03] BAUER J., KARNER K., SCHINDLER K., KLAUS A., ZACH C.: Segmentation of building models from dense 3D point-clouds. In *27th Workshop of the Austrian Association for Pattern Recognition* (2003). 28, 30

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (1992), 239–256. 95

[BM10] BALDAUF M., MUSIALSKI P.: A Device-aware Spatial 3D Visualization Platform for Mobile Urban Exploration. In *The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2010)* (Florence, Italy, 2010), IARIA. 5

[BR06] BRENNER C., RIPPERDA N.: Extraction of Facades using rjMCMC and Constraint Equations. In *PCV '06, Photogrammetric Computer Vision* (Bonn, 2006), ISPRS Comm. III Symposium, IAPRS, pp. 155–160. 25

[BTP09] BUROCHIN J.-P., TOURNAIRE O., PAPARODITIS N.: An unsupervised hierarchical segmentation of a facade building image in elementary 2d - models. In *ISPRS Workshop, CMRT09 - City Models, Roads and Traffic* (2009). 13

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239. 72

[BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *ACM Transactions on Graphics 29*, 4 (July 2010), 1. 23

[BZB06] BAUER J., ZACH C., BISCHOF H.: Efficient Sparse 3D Reconstruction by Space Sweeping. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)* (June 2006), IEEE, pp. 527–534. 30

[Can86] CANNY J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (Nov. 1986), 679–698. 55

[CKX*08] CHEN X., KANG S. B., XU Y.-Q., DORSEY J., SHUM H.-Y.: Sketching reality. *ACM Transactions on Graphics 27*, 2 (Apr. 2008), 1–15. 21

[CLCvG07] CORNELIS N., LEIBE B., CORNELIS K., VAN GOOL L.: 3D Urban Scene Modeling Integrating Recognition and Reconstruction. *International Journal of Computer Vision 78*, 2-3 (Oct. 2007), 121–141. 27

[CM02] COMANICIU D., MEER P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 603–619. 50, 57, 82

[Col96] COLLINS R. T.: A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1996), IEEE Comput. Soc. Press, pp. 358–363. 30

[CR99] CIPOLLA R., ROBERTSON D.: 3D models of architectural scenes from uncalibrated images and vanishing points. In *Proceedings 10th International Conference on Image Analysis and Processing* (1999), vol. 0, IEEE Comput. Soc, pp. 824–829. 17

[CS08] CECH J., SARA R.: Windowpane Detection based on Maximum Aposteriori Probability Labeling. In *Image Analysis - From Theory to Applications* (2008). 14

[CT99] COORG S., TELLER S.: Extracting textured vertical facades from controlled close-range imagery. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)* (1999), IEEE Comput. Soc, pp. 625–632. 19

[CZM*10] CHENG M.-M., ZHANG F.-L., MITRA N. J., HUANG X., HU S.-M.: RepFinder: finding approximately repeated scene elements for image editing. *ACM Transactions on Graphics 29*, 4 (July 2010), 1. 90

[DF08] DRAUSCHKE M., FÖRSTNER W.: Selecting appropriate features for detecting buildings and building parts. In *21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), Beijing, China* (2008). 14

[DFKE07] DABOV K., FOI A., KATKOVNIK V., EGIAZARIAN K.: Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing 16*, 8 (Aug. 2007), 2080–2095. 80

[DTC04] DICK A., TORR P. H. S., CIPOLLA R.: Modelling and Interpretation of Architecture from Several Images. *International Journal of Computer Vision 60*, 2 (Nov. 2004), 111–134. 24, 25

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and rendering architecture from photographs. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96* (New York, New York, USA, 1996), ACM Press, pp. 11–20. 16, 19

[DYB98] DEBEVEC P. E., YU Y., BORSHUKOV G.: Efficient view-dependent image-based rendering with projective texture-mapping. In *Rendering techniques' 98: proceedings of the Eurographics Workshop in Vienna, Austria, June 29-July 1, 1998* (1998), Springer Verlag Wien, p. 105. 20

[EDM*08] EISEMANN M., DE DECKER B., MAGNOR M. A., BEKAERT P., DE AGUIAR E., AHMED N., THEOBALT C., SELLENT A.: Floating Textures. *Computer Graphics Forum 27*, 2 (Apr. 2008), 409–418. 20

[EhWGG05] EL-HAKIM S., WHITING E., GONZO L., GIRARDI S.: 3D Reconstruction of Complex Architectures from Multiple Data. In *Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005: "Virtual Reconstruction and Visualization of Complex Architectures"* (Mestre-Venice, Italy, 2005). 30

[ELS08] EISENACHER C., LEFEBVRE S., STAMMINGER M.: Texture Synthesis From Photographs. *Computer Graphics Forum 27*, 2 (Apr. 2008), 419–428. 20

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (June 1981), 381–395. 15, 36

[FCSS09a] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (June 2009), IEEE, pp. 1422–1429. 28

[FCSS09b] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Reconstructing Building Interiors from Images. In *2009 IEEE 12th International Conference on Computer Vision* (2009), IEEE. 28

[FCSS10] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Towards Internet-scale Multi-view Stereo. In *2010 IEEE Conference on Computer Vision and Pattern Recognition* (2010), IEEE, p. to appear. 28

[Fin08] FINKENZELLER D.: Detailed Building Facades. *IEEE Computer Graphics and Applications 28*, 3 (May 2008), 58–66. 22

[FJZ05] FRUEH C., JAIN S., ZAKHOR A.: Data Processing Algorithms for Generating Textured 3D Building Facade Meshes from Laser Scans and Camera Images. *International Journal of Computer Vision 61*, 2 (Feb. 2005), 159–184. 29

[FP07] FURUKAWA Y., PONCE J.: Accurate, Dense, and Robust Multi-View Stereopsis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE, pp. 1–8. 28, 32

[FP09] FURUKAWA Y., PONCE J.: Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99 (2009), 1–1. 28

[FZ03] FRUEH C., ZAKHOR A.: Constructing 3D city models by merging aerial and ground views. *IEEE Computer Graphics and Applications 23*, 6 (Nov. 2003), 52–61. 29

[FZ04] FRUEH C., ZAKHOR A.: An Automated Method for Large-Scale, Ground-Based City Model Acquisition. *International Journal of Computer Vision 60*, 1 (Oct. 2004), 5–24. 29

[GCS06] GOESELE M., CURLESS B., SEITZ S.: Multi-View Stereo Revisited. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)* (2006), IEEE, pp. 2402–2409. 31

[GFM*07] GALLUP D., FRAHM J.-M., MORDOHAI P., YANG Q., POLLEFEYS M.: Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE, pp. 1–8. 30

[GG84] GEMAN S., GEMAN D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, 6 (Nov. 1984), 721–741. 26

[GH97] GUPTA R., HARTLEY R.: Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 9 (1997), 963–975. 9

[GKVH09] GRZESZCZUK R., KOŠECKÁ J., VEDANTHAM R., HILE H.: Creating compact architectural models by geo-registering image collections. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (Sept. 2009), IEEE, pp. 1718–1725. 30

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S. M.: Multi-View Stereo for Community Photo Collections. In *2007 IEEE 11th International Conference on Computer Vision* (Oct. 2007), IEEE, pp. 1–8. 16

[GSGA05] GEORGIADIS C., STEFANIDIS A., GYFTAKIS S., AGOURIS P.: Image Orientation for Interactive Tours of Virtually-Modeled Sites. In *Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005: "Virtual Reconstruction and Visualization of Complex Architectures"* (Mestre-Venice, Italy, 2005). 30

[GWOH10] GAL R., WEXLER Y., OFEK E., HOPPE H.: Seamless Montage for Texturing Models. *Computer Graphics Forum 29*, 2 (2010), to appear. 19, 20

[Hav05] HAVEMANN S.: *Generative Mesh Modeling*. Phd, Technische Universität Braunschweig, 2005. 26

[HKHF09] HOHMANN B., KRISPEL U., HAVEMANN S., FELLNER D.: CityFit - High-quality urban reconstructions by fitting shape grammars to images and derived textured point clouds. In *Proceedings of the 3rd ISPRS International Workshop 3D-ARCH 2009: "3D Virtual Reconstruction and Visualization of Complex Architectures"* (Trento, Italy, 25-28 February 2009, 2009). 25

[HLEL06] HAYS J., LEORDEANU M., EFROS A. A., LIU Y.: Discovering Texture Regularity as a Higher-Order Correspondence Problem. In *Computer Vision - ECCV 2006* (2006), Leonardis A., Bischof H., Pinz A., (Eds.), vol. 3952, Springer Berlin Heidelberg, pp. 522–535. 11

[HLL01] HSU J. T., LIU L.-C., LI C.: Determination of structure component in image texture using wavelet analysis. In *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)* (2001), IEEE, pp. 166–169. 11

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A 4*, 4 (Apr. 1987), 629. 95, 96

[HPFP09] HAUGEARD J.-E., PHILIPP-FOLIGUET S., PRECIOSO F.: Windows and facades retrieval using similarity on graph of contours. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (Nov. 2009), IEEE, pp. 269–272. 14

[HS88] HARRIS C., STEPHENS M.: A Combined Corner and Edge Detector. In *Alvey vision conference* (Manchester, 1988), pp. 147–151. 49

[HTF09] HASTIE T., TIBSHIRANI R., FRIEDMAN J. H.: *The elements of statistical learning: data mining, inference, and prediction*, 2 ed. Springer, 2009. 12, 67, 69, 70, 89

[HYN03] HU J., YOU S., NEUMANN U.: Approaches to large-scale urban modeling. *IEEE Computer Graphics and Applications 23*, 6 (Nov. 2003), 62–69. 27

[HZ04] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*. {Cambridge University Press}, Mar. 2004. 15

[HZ05] HAN F., ZHU S.-C.: Bottom-up/Top-Down Image Parsing by Attribute Graph Grammar. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (2005), vol. 2, IEEE, pp. 1778–1785. 24

[HZ09] HAN F., ZHU S.-C.: Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 1 (Jan. 2009), 59–73. 2, 24

[IZB07] IRSCHARA A., ZACH C., BISCHOF H.: Towards Wiki-based Dense City Modeling. In *2007 IEEE 11th International Conference on Computer Vision* (Oct. 2007), IEEE, pp. 1–8. 16, 27, 31, 35

[JCW09] JESCHKE S., CLINE D., WONKA P.: A GPU Laplacian solver for diffusion curves and Poisson image editing. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), 1. 10

[JRH03] JAYNES C., RISEMAN E., HANSON A. R.: Recognition and reconstruction of buildings from multiple aerial images. *Computer Vision and Image Understanding 90*, 1 (Apr. 2003), 68–98. 30

[JTC09] JIANG N., TAN P., CHEONG L.-F.: Symmetric architecture modeling with a single image. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), 1. 19

[KNC*08] KOPF J., NEUBERT B., CHEN B., COHEN M., COHEN-OR D., DEUSSEN O., UYTTENDAELE M., LISCHINSKI D.: Deep Photo: Model-Based Photograph Enhancement and Viewing. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1. 20

[Kol06] KOLLER M.: Seamless City. **http://www.seamlesscity.com**, 2006. 9

[KP10] KARANTZALOS K., PARAGIOS N.: Large-Scale Building Reconstruction Through Information Fusion and 3-D Priors. *IEEE Transactions on Geoscience and Remote Sensing 48*, 5 (May 2010), 2283–2296. 30

[KR07a] KORAH T., RASMUSSEN C.: 2D Lattice Extraction from Structured Environments. In *2007 IEEE International Conference on Image Processing* (2007), vol. 2, IEEE, pp. II – 61–II – 64. 25

[KR07b] KORAH T., RASMUSSEN C.: Spatiotemporal Inpainting for Recovering Texture Maps of Occluded Building Facades. *IEEE Transactions on Image Processing 16*, 9 (Sept. 2007), 2262–2271. 20

[KST*09] KOUTSOURAKIS P., SIMON L., TEBOUL O., TZIRITAS G., PARAGIOS N.: Single view reconstruction using shape grammars for urban environments. In *2009 IEEE 12th International Conference on Computer Vision* (Sept. 2009), IEEE, pp. 1795–1802. 26

[KZ04] KOLMOGOROV V., ZABIH R.: What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (Feb. 2004), 147–59. 72

[KZ05] KOŠECKÁ J., ZHANG W.: Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding 100*, 3 (Dec. 2005), 274–293. 28

[KZZL10] KANG Z., ZHANG L., ZLATANOVA S., LI J.: An automatic mosaicking method for building facade texture mapping using a monocular close-range image sequence. *ISPRS Journal of Photogrammetry and Remote Sensing 65*, 3 (May 2010), 282–293. 20

[LA04] LOURAKIS M. I. A., ARGYROS A. A.: The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm, 2004. 15

[LBHL08] LIU Y., BELKINA T., HAYS J., LUBLINERMAN R.: Image de-fencing. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), IEEE, pp. 1–8. 11

[LCT04] LIU Y., COLLINS R. T., TSIN Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 3 (Mar. 2004), 354–71. 11

[LCZ99] LIEBOWITZ D., CRIMINISI A., ZISSERMAN A.: Creating Architectural Models from Images. *Computer Graphics Forum 18*, 3 (Sept. 1999), 39–50. 17

[LDZPD10] LAFARGE F., DESCOMBES X., ZERUBIA J., PIERROT-DESEILLIGNY M.: Structural approach for building reconstruction from a single DSM. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 1 (Jan. 2010), 135–47. 30

[LE06] LOY G., EKLUNDH J.-O.: Detecting Symmetry and Symmetric Constellations of Features. In *Computer Vision - ECCV 2006* (Berlin, Heidelberg, 2006), Leonardis A., Bischof H., Pinz A., (Eds.), vol. 3952 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 508–521. 11

[LHXS05] LIU Y., HAYS J., XU Y.-Q., SHUM H.-Y.: Digital papercutting. In *ACM SIGGRAPH 2005 Sketches on - SIGGRAPH '05* (New York, New York, USA, 2005), ACM Press, p. 99. 11

[LI07] LEMPITSKY V., IVANOV D.: Seamless Mosaicing of Image-Based Texture Maps. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE, pp. 1–6. 20

[LLH04] LIU Y., LIN W.-C., HAYS J.: Near-regular texture analysis and manipulation. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 368. 11

[LMWY09] LIU J., MUSIALSKI P., WONKA P., YE J.: Tensor completion for estimating missing values in visual data. In *2009 IEEE 12th International Conference on Computer Vision* (Sept. 2009), IEEE, pp. 2114–2121. 5, 12

[LN03] LEE S. C., NEVATIA R.: Interactive 3D building modeling using a hierarchical representation. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003.* (2003), IEEE Comput. Soc, pp. 58–65. 17

[LN04] LEE S. C., NEVATIA R.: Extraction and integration of window in a 3D building model from ground view images. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (2004), vol. 2, IEEE, pp. 113–120. 13

[Low04] LOWE D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*, 2 (Nov. 2004), 91–110. 15, 35

[Lux07] LUXBURG U.: A tutorial on spectral clustering. *Statistics and Computing 17*, 4 (Aug. 2007), 395–416. 71

[LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 22

[LZ98] LIEBOWITZ D., ZISSERMAN A.: Metric rectification for perspective images of planes. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1998), IEEE Comput. Soc, pp. 482–488. 17, 19

[MAW*07] MERRELL P., AKBARZADEH A., WANG L., MORDOHAI P., FRAHM J.-M., YANG R., NISTER D., POLLEFEYS M.: Real-Time Visibility-Based Fusion of Depth Maps. In *2007 IEEE 11th International Conference on Computer Vision* (Oct. 2007), IEEE, pp. 1–8. 31

[May08] MAYER H.: Object extraction in photogrammetric computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing 63*, 2 (Mar. 2008), 213–222. 25

[Mer07] MERRELL P.: Example-based model synthesis. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games - I3D '07* (New York, New York, USA, 2007), ACM Press, p. 105. 22

[Mey78] MEYER F.: Contrast feature extraction. In *Quantitative Analysis of Microstructures in Material Sciences, Biology and Medicine* (Stuttgart, FRG, 1978), Chermant J.-L., (Ed.), Riederer Verlag. 68

[MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics 25*, 3 (July 2006), 560. 11

[MGP07] MITRA N. J., GUIBAS L. J., PAULY M.: Symmetrization. *ACM Transactions on Graphics 26*, 3 (July 2007), 63. 11

[MK10] MIČUŠÍK B., KOŠECKÁ J.: Multi-view Superpixel Stereo in Urban Environments. *International Journal of Computer Vision 89*, 1 (Mar. 2010), 106–119. 31, 32

[MKF09] MASTIN A., KEPNER J., FISHER J.: Automatic registration of LIDAR and optical images of urban scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL, June 2009), IEEE, pp. 2639–2646. 29

[Mül10] MÜLLER P.: Procedural Inc. **http://www.procedural.com**, October 2010. 21

[MLS*10] MUSIALSKI P., LUKSCH C., SCHWÄRZLER M., BUCHETICS M., MAIERHOFER S., PURGATHOFER W.: Interactive Multi-View Façade Image Editing. In *Vision, Modeling, Visualisation (VMV'10)* (2010). 4, 5, 20

[MM08] MERRELL P., MANOCHA D.: Continuous model synthesis. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1. 22

[MM09] MERRELL P., MANOCHA D.: Constraint-based model synthesis. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling on - SPM '09* (New York, New York, USA, 2009), ACM Press, p. 101. 22

[MP08] MCCANN J., POLLARD N. S.: Real-time gradient-domain painting. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 10, 40

[MPB05] MARVIE J.-E., PERRET J., BOUATOUCH K.: The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer 21*, 5 (May 2005), 329–339. 22

[MR05] MAYER H., REZNIK S.: Building Façade Interpretation from Image Sequences. In *Proceedings of the ISPRS Workshop CMRT 2005* (Vienna, 2005), vol. XXXVI, pp. 55–60. 25

[MR06] MAYER H., REZNIK S.: MCMC Linked with Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences. In *PCV '06, Photogrammetric Computer Vision* (2006), ISPRS Comm. III Symposium, IAPRS, pp. 130–135. 25

[MR07] MAYER H., REZNIK S.: Building facade interpretation from uncalibrated wide-baseline image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing 61*, 6 (Feb. 2007), 371–380. 13, 25

[MRM*10] MUSIALSKI P., RECHEIS M., MAIERHOFER S., WONKA P., PURGATHOFER W.: Tiling of Ortho-Rectified Façade Images. In *Spring Conference on Computer Graphics (SCCG'10)* (Budmerice, 2010). 4, 5, 33

[MS01] MEILA M., SHI J.: A Random Walks View of Spectral Segmentation. In *Proc. Int. Conf. Artificial Intelligence and Statistics* (2001). 71

[Mus09] MUSIALSKI P.: *Point Cloud to Model Registration*. Tech. rep., VRVis Research Center, Vienna, July 2009. 5

[Mus10] MUSIALSKI P.: *Axis-Aligned Segmentation of Orthographic Façade Images*. Tech. rep., VRVis Research Center, Vienna, September 2010. 4, 5

[MVW*06] MÜLLER P., VEREENOOGHE T., WONKA P., PAAP I., VAN GOOL L.: Procedural 3D Reconstruction of Puuc Buildings in Xkipché. In *The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, VAST (2006)* (2006), EG, pp. 139–146. 21

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Transactions on Graphics 25*, 3 (July 2006), 614. 21, 22

[MWR*09] MUSIALSKI P., WONKA P., RECHEIS M., MAIERHOFER S., PURGATHOFER W.: Symmetry-Based Façade Repair. In *Vision, Modeling, Visualisation (VMV'09)* (2009), Magnor M. A., Rosenhahn B., Theisel H., (Eds.), DNB, pp. 3–10. 4, 5, 20, 33

[MZWvG07] MÜLLER P., ZENG G., WONKA P., VAN GOOL L.: Image-based procedural modeling of facades. *ACM Transactions on Graphics 26*, 3 (July 2007), 85. 14, 25, 26, 33

[Nis04] NISTÉR D.: An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 6 (June 2004), 756–77. 15

[NJW01] NG A., JORDAN M., WEISS Y.: On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems* (2001), Dietterich T., Becker S., Ghahramani Z., (Eds.), MIT Press, pp. 849–856. 71

[NK01] NEUGEBAUER P. J., KLEIN K.: Texturing 3D Models of Real World Objects from Multiple Unregistered Photographic Views. *Computer Graphics Forum 18*, 3 (Sept. 2001), 245–256. 20

[PDG05] PORQUET D., DISCHLER J.-M., GHAZANFARPOUR D.: Real-time high-quality View-Dependent Texture Mapping using per-pixel visibility. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia - GRAPHITE '05* (New York, New York, USA, 2005), ACM Press, p. 213. 20

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics 22*, 3 (July 2003), 313. 10, 39

[PHT93] PRUSINKIEWICZ P., HAMMEL M., TN C.: A Fractal Model of Mountains with Rivers A Fractal Model of Mountains with Rivers. In *Proceeding of Graphics Interface '93* (1993), no. May, pp. 174–180. 20

[PHYH06] POTTMANN H., HUANG Q.-X., YANG Y.-L., HU S.-M.: Geometry and Convergence Analysis of Algorithms for Registration of 3D Shapes. *International Journal of Computer Vision 67*, 3 (Mar. 2006), 277–296. 98

[PL90] PRUSINKIEWICZ P., LINDENMAYER A.: *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, 1990. 20

[PLH04] POTTMANN H., LEOPOLDSEDER S., HOFER M.: Registration without ICP. *Computer Vision and Image Understanding 95*, 1 (July 2004), 54–71. 98, 99

[PM90] PERONA P., MALIK J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (July 1990), 629–639. 79

[PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01* (New York, New York, USA, 2001), ACM Press, pp. 301–308. 21

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L. J.: Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 11

[PNF*07] POLLEFEYS M., NISTÉR D., FRAHM J.-M., AKBARZADEH A., MORDOHAI P., CLIPP B., ENGELS C., GALLUP D., KIM S.-J., MERRELL P., SALMI C., SINHA S. N.,

TALTON B., WANG L., YANG Q., STEWÉNIUS H., YANG R., WELCH G., TOWLES H.: Detailed Real-Time Urban 3D Reconstruction from Video. *International Journal of Computer Vision 78*, 2-3 (Oct. 2007), 143–167. 28, 30

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics 25*, 3 (July 2006), 549. 11

[PSK06] PAVIĆ D., SCHÖNEFELD V., KOBBELT L.: Interactive image completion with perspective correction. *The Visual Computer 22*, 9-11 (Aug. 2006), 671–681. 20

[PV09a] PU S., VOSSELMAN G.: Building Facade Reconstruction by Fusing Terrestrial Laser Points and Images. *Sensors 9*, 6 (June 2009), 4525–4542. 26, 29

[PV09b] PU S., VOSSELMAN G.: Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing 64*, 6 (Nov. 2009), 575–584. 29

[PV09c] PU S., VOSSELMAN G.: Refining building facade models with images. In *ISPRS Workshop, CMRT09 - City Models, Roads and Traffic* (2009), vol. XXXVIII, pp. 3–4. 26, 29

[PvGV*04] POLLEFEYS M., VAN GOOL L., VERGAUWEN M., VERBIEST F., CORNELIS K., TOPS J., KOCH R.: Visual Modeling with a Hand-Held Camera. *International Journal of Computer Vision 59*, 3 (Sept. 2004), 207–232. 27, 30, 35

[PY09] POULLIS C., YOU S.: Photorealistic large-scale urban city model reconstruction. *IEEE Transactions on Visualization and Computer Graphics 15*, 4 (2009), 654–69. 30

[RAGS01] REINHARD E., ADHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications 21*, 4 (2001), 34–41. 40

[RB07] RIPPERDA N., BRENNER C.: Data driven rule proposal for grammar based façade reconstruction. In *PIA07. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2007). 25

[RB09] RIPPERDA N., BRENNER C.: Application of a Formal Grammar to Facade Reconstruction in Semiautomatic and Automatic Environments. In *Proceedings of 12th AGILE Conference on GIScience* (Hannover, Germany, 2009). 25

[RC02] ROTHER C., CARLSSON S.: Linear Multi View Reconstruction and Camera Recovery Using a Reference Plane. *International Journal of Computer Vision 49*, 2 (2002), 117–141–141. 28

[RGL04] ROMAN A., GARG G., LEVOY M.: Interactive design of multi-perspective images for visualizing urban landscapes. *IEEE Visualization 2004* (2004), 537–544. 9, 10

[Rip08] RIPPERDA N.: Determination of Facade Attributes for Facade Reconstruction. In *ISPRS Congress Beijing 2008, Proceedings of Commission III* (2008), pp. 285–290. 25

[RM07] REZNIK S., MAYER H.: Implicit shape models, model selection, and plane sweeping for 3d facade interpretation. In *PIA07. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2007), vol. 36, pp. 173–178. 25

[RWY95] REISFELD D., WOLFSON H., YESHURUN Y.: Context-free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision 14*, 2 (Mar. 1995), 119–130. 11

[SA00] STAMOS I., ALLEN P. K.: 3-D model construction using range and image data. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000* (2000), IEEE Comput. Soc, pp. 531–536. 28

[SA01] STAMOS I., ALLEN P. K.: Automatic registration of 2-D with 3-D imagery in urban environments. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (2001), IEEE Comput. Soc, pp. 731–736. 28

[SA02] STAMOS I., ALLEN P. K.: Geometry and Texture Recovery of Scenes of Large Scale. *Computer Vision and Image Understanding 88*, 2 (Nov. 2002), 94–118. 19, 28

[SB03] SCHINDLER K., BAUER J.: A model-based method for building reconstruction. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003.* (2003), IEEE Comput. Soc, pp. 74–82. 13, 28

[SBM*10] ST'AVA O., BENEŠ B., MECH R., ALIAGA D. G., KRIŠTOF P.: Inverse Procedural Modeling by Automatic Generation of L-systems. *Computer Graphics Forum 29*, 2 (2010). 24

[SGSS08] SNAVELY N., GARG R., SEITZ S. M., SZELISKI R.: Finding paths through the world's photos. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 16, 20

[SHS98] SHUM H.-Y., HAN M., SZELISKI R.: Interactive construction of 3D models from panoramic mosaics. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1998), IEEE Comput. Soc, pp. 427–433. 19

[SI07] SHECHTMAN E., IRANI M.: Matching Local Self-Similarities across Images and Videos. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (Minneapolis, MN, USA, June 2007), IEEE, pp. 1–8. 11

[Sip96] SIPSER M.: *Introduction to the Theory of Computation*. PWS Pub. Co., 1996. 20

[SK03] SEITZ S. M., KIM J.: Multiperspective imaging. *IEEE Computer Graphics and Applications 23*, 6 (Nov. 2003), 16–19. 9

[SKD06] SCHINDLER G., KRISHNAMURTHY P., DELLAERT F.: Line-Based Structure from Motion for Urban Environments. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)* (June 2006), IEEE, pp. 846–853. 15

[SS99] SCHAPIRE R. E., SINGER Y.: Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning 37*, 3 (1999), 297. 14

[SS10] SONG Y., SHAN J.: Color correction of texture images for true photorealistic visualization. *ISPRS Journal of Photogrammetry and Remote Sensing 65*, 3 (May 2010), 308–315. 40

[SSG*10] SNAVELY N., SIMON I., GOESELE M., SZELISKI R., SEITZ S. M.: Scene Reconstruction and Visualization From Community Photo Collections. *Proceedings of the IEEE 98*, 8 (Aug. 2010), 1370–1390. 14

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics 25*, 3 (July 2006), 835. 16, 20

[SSS07] SNAVELY N., SEITZ S. M., SZELISKI R.: Modeling the World from Internet Photo Collections. *International Journal of Computer Vision 80*, 2 (Dec. 2007), 189–210. 16, 20

[SSS*08] SINHA S. N., STEEDLY D., SZELISKI R., AGRAWALA M., POLLEFEYS M.: Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1. 16, 18, 20

[Sti75] STINY G. N.: Pictorial and formal aspects of shape and shape grammars and aesthetic systems. 417. 20

[Str05] STRANG G.: *Linear Algebra and Its Applications*. Brooks Cole, 2005. 12

[Sze06] SZELISKI R.: Image Alignment and Stitching: A Tutorial. *Foundations and Trends in Computer Graphics and Vision 2*, 1 (2006), 1–104. 9

[TBTS08] TAI Y.-W., BROWN M. S., TANG C.-K., SHUM H.-Y.: Texture amendment. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1. 20

[TCLH06] TSAI F., CHEN C.-H., LIU J.-K., HSIAO K.-H.: Texture Generation and Mapping Using Video Sequences for 3D Building Models. In *Innovations in 3D Geo Information Systems* (Berlin, Heidelberg, 2006), Abdul-Rahman A., Zlatanova S., Coors V., (Eds.), Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, pp. 429–438. 13

[Tel98] TELLER S.: Automated Urban Model Acquisition: Project Rationale and Status. In *Image Understanding Workshop* (1998), pp. 445–462. 28

[TKO08] TAN Y. K. A., KWOH L. K., ONG S. H.: Large scale texture mapping of building facades. In *ISPRS Congress Beijing 2008, Proceedings of Commission V* (2008). 20

[TLH06] TSAI F., LIU J.-K., HSIAO K.-H.: Morphological Processing of Video for 3D Building Model Visualization. In *Proceedings of 27 th Asian Conference on Remote Sensing (ACRS2006), Ulaanbaatar, Mongolia* (2006), pp. 1–6. 13

[TLLH05] TSAI F., LIN H.-C., LIU J.-K., HSIAO K.-H.: Semiautomatic texture generation and transformation for cyber city building models. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International* (2005), pp. 4980–4983. 13

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision* (1998), Narosa Publishing House, pp. 839–846. 79

[TMHF99] TRIGGS B., MCLAUCHLAN P. F., HARTLEY R. I., FITZGIBBON A. W.: Bundle Adjustment - A Modern Synthesis. *Lecture Notes In Computer Science; Vol. 1883* (1999), 298. 15

[TS08] THORMÄHLEN T., SEIDEL H. P.: 3D-modeling by ortho-image generation from image sequences. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 20

[TSKP10] TEBOUL O., SIMON L., KOUTSOURAKIS P., PARAGIOS N.: Segmentation of building facades using procedural shape priors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 2010), IEEE, pp. 3105–3112. 26

[TTMvG01] TURINA A., TUYTELAARS T., MOONS T., VAN GOOL L.: Grouping via the Matching of Repeated Patterns. Singh S., Murshed N., Kropatsch W., (Eds.), Lecture Notes in Computer Science, Springer, pp. 250–259. 11

[TTvG01] TURINA A., TUYTELAARS T., VAN GOOL L.: Efficient grouping under perspective skew. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (2001), vol. 1, IEEE Comput. Soc, pp. I–247–I–254. 11

[VAB10] VANEGAS C. A., ALIAGA D. G., BENEŠ B.: Building Reconstruction using Manhattan-World Grammars. *2010 IEEE Conference on Computer Vision and Pattern Recognition* (2010), to appear. 24, 29, 30

[VABW09] VANEGAS C. A., ALIAGA D. G., BENEŠ B., WADDELL P. A.: Interactive design of urban spaces using geometrical and behavioral modeling. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), 1. 21

[VAW*10] VANEGAS C. A., ALIAGA D. G., WONKA P., MÜLLER P., WADDELL P. A., WATSON B.: Modelling the Appearance and Behaviour of Urban Spaces. *Computer Graphics Forum 29*, 1 (Mar. 2010), 25–42. 20, 23

[vdHDT*06] VAN DEN HENGEL A., DICK A., THORMÄHLEN T., WARD B., TORR P. H. S.: Building models of regular scenes from structure and motion. *British Machine Vision Conference 2006* (2006). 17

[vdHDT*07a] VAN DEN HENGEL A., DICK A., THORMÄHLEN T., WARD B., TORR P. H. S.: A shape hierarchy for 3D modelling from video. *Computer graphics and interactive techniques in Australasia and South East Asia* (2007), 63. 17, 18

[vdHDT*07b] VAN DEN HENGEL A., DICK A., THORMÄHLEN T., WARD B., TORR P. H. S.: Interactive 3D Model Completion. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)* (Dec. 2007), IEEE, pp. 175–181. 18

[vdHDT*07c] VAN DEN HENGEL A., DICK A., THORMÄHLEN T., WARD B., TORR P. H. S.: VideoTrace: rapid interactive scene modelling from video. *ACM Transactions on Graphics 26*, 3 (July 2007), 86. 20

[vGZBM07] VAN GOOL L., ZENG G., BORRE F. V. D., MÜLLER P.: Towards Mass-produced Building Models. In *PIA07. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2007), Stilla U., Mayer H., Rottensteiner F., Heipke C., Hinz S., (Eds.), Institute of Photogrammetry and Cartography, Technische Universitaet Muenchen, pp. 209–220. 19, 25

[VvG06] VERGAUWEN M., VAN GOOL L.: Web-based 3D Reconstruction Service. *Machine Vision and Applications 17*, 6 (2006), 411. 16, 30, 32

[WMV*08] WATSON B., MÜLLER P., VERYOVKA O., FULLER A., WONKA P., SEXTON C.: Procedural Urban Modeling in Practice. *IEEE Computer Graphics and Applications 28*, 3 (May 2008), 18–26. 21

[WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), 1. 22

[WTT*02] WANG X., TOTARO S., TAILL F., HANSON A. R., TELLER S.: Recovering facade texture and microstructure from real-world images. In *Photogrammetric Computer Vision, ISPRS Commission III, Symposium 2002* (Graz, Austria, 2002), no. September, pp. A381–368. 13

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Transactions on Graphics 22*, 3 (July 2003), 669. 20, 21, 22, 23

[WYN07] WANG L., YOU S., NEUMANN U.: Semiautomatic registration between ground-level panoramas and an orthorectified aerial image for building modeling. In *2007 IEEE 11th International Conference on Computer Vision* (2007), IEEE, pp. 1–8. 29

[WZ02] WERNER T., ZISSERMAN A.: New Techniques for Automated Architectural Reconstruction from Photographs. *Lecture Notes In Computer Science; Vol. 2351* (2002). 17

[WZ08] WILSON R. C., ZHU P.: A study of graph spectra for comparing graphs and trees. *Pattern Recognition 41*, 9 (Sept. 2008), 2833–2841. 71

[XFT*08] XIAO J., FANG T., TAN P., ZHAO P., OFEK E., QUAN L.: Image-based façade modeling. *ACM Transactions on Graphics 27*, 5 (Dec. 2008), 1. 18, 20, 28

[XFZ*09] XIAO J., FANG T., ZHAO P., LHUILLIER M., QUAN L.: Image-based street-side city modeling. *ACM Transactions on Graphics (TOG) 28*, 5 (2009). 20, 28

[YP03] YANG R., POLLEFEYS M.: Multi-resolution real-time stereo on commodity graphics hardware. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* (2003), IEEE Comput. Soc, pp. I–211–I–217. 30

[YWR09] YIN X., WONKA P., RAZDAN A.: Generating 3D Building Models from Architectural Drawings: A Survey. *IEEE Computer Graphics and Applications 29*, 1 (Jan. 2009), 20–30. 7

[ZBKB08] ZEBEDIN L., BAUER J., KARNER K., BISCHOF H.: Fusion of Feature- and Area-Based Information for Urban Buildings Modeling from Aerial Imagery. In *Computer Vision - ECCV 2008* (Berlin, Heidelberg, 2008), Forsyth D., Torr P. H. S., Zisserman A., (Eds.), vol. 5305 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 873–886. 30

[ZFPW03] ZOMET A., FELDMAN D., PELEG S., WEINSHALL D.: Mosaicing new views: the crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 6 (June 2003), 741–754. 9

[Zhe03] ZHENG J. Y.: Digital route panoramas. *IEEE Multimedia 10*, 3 (July 2003), 57–67. 9

[ZLPW06] ZOMET A., LEVIN A., PELEG S., WEISS Y.: Seamless image stitching by minimizing false edges. *IEEE Transactions on Image Processing 15*, 4 (Apr. 2006), 969–977. 10

[ZPB07] ZACH C., POCK T., BISCHOF H.: A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration. In *2007 IEEE 11th International Conference on Computer Vision* (Oct. 2007), IEEE, pp. 1–8. 31

[ZSW*10] ZHENG Q., SHARF A., WAN G., LI Y., MITRA N. J., COHEN-OR D., CHEN B.: Non-local scan consolidation for 3D urban scenes. *ACM Transactions on Graphics 29*, 4 (July 2010), 1. 90

# Curriculum Vitae

## Personal

| | |
|---|---|
| **Name Surname** | Przemyslaw Musialski |
| **Gender** | male |
| **Day of Birth** | May 18th, 1978 in Olsztyn, Poland |
| **Marital Status** | married |
| **Academic Degree** | Dipl.-Mediensystemwissenschaftler (equ. M.Sc.) |
| **Languages** | German, English, Polish |

## Education

| | |
|---|---|
| **since 2007/10** | Ph.D. student in Computer Science, TU-Vienna, Austria<br>Advisors: Prof. Dr. Werner Purgathofer and Prof. Dr. Peter Wonka |
| **2007/09** | Dipl.-Mediensystemwiss. , Bauhaus-University Weimar, Germany<br>Advisors: Prof. Dr. Charles A. Wüthrich and Dr. Robert F. Tobler |
| **1999/06** | Abitur (A-Levels), Clavius Gymnasium Bamberg, Germany |

## Professional

| | |
|---|---|
| **2010/07 – present** | Project Assistant; Institute of Computer Graphics and Algorithms, TU-Vienna |
| **2007/06 – present** | Researcher; VRVis Research Center, Vienna, Austria |
| **2006/04 – 2007/05** | Internship; VRVis Research Center, Vienna, Austria |
| **2005/10 – 2006/03** | Student Assistant; Computer Graphics Group, Bauhaus-University Weimar |
| **2003/10 – 2004/03** | Student Assistant; Computer Graphics Group, Bauhaus-University Weimar |

## Abroad Experience

| | |
|---|---|
| **2004/09 – 2005/05** | University of the Philippines Diliman, Quezon City, Philippines<br>Graduate Exchange Student at the College of Engineering<br>Department of Computer Science |

## Collaborators and scientific Advisors

Werner Purgathofer, Vienna University of Technology, Austria (PhD Advisor)
Peter Wonka, Arizona State University, USA (PhD Co-Advisor)
Charles A. Wüthrich, Bauhaus-University Weimar, Germany (Diploma Advisor)
Robert F. Tobler, VRVis Research Center, Vienna, Austria (Diploma Co-Advisor)
Gordon Wetzstein, University of British Columbia, Canada (Collaborator)
Matthias Baldauf, Forschungszentrum Telekommunikation Wien, Austria (Collaborator)
Peter Fröhlich, Forschungszentrum Telekommunikation Wien, Austria (Collaborator)

## Reviewing

Web3D (IPC Member)
ACM SIGGRAPH 2009, 2010
Eurographics 2010
Computer Graphics International (CGI 2009)
3DPVT 2010
WSCG 2010

## Talks

| | |
|---|---|
| **2009/11/25** | Symmetry-Based Facade Repair<br>VRVis Forum, Vienna, Austria |
| **2009/07/21** | Image Based Reconstruction of an Urban Environments<br>Silesian University in Opava, Czech Republic (Invited Talk) |
| **2008/07/19** | Collaborative Reconstruction of Urban Environments from Ground-Based Images<br>Silesian University in Opava, Czech Republic (Invited Talk) |
| **2007/11/23** | Multiresolution Geometric Details on Subdivison Surfaces<br>CG-Konversatorium, Vienna University of Technology, Austria |

## Supervised Students

| | |
|---|---|
| **2008/10 – 2009/12** | Meinrad Recheis, Master Thesis<br>Automatic Recognition of Repeating Patterns in Rectified Facade Images. |
| **2009/06 – 2009/09** | Micheal Hornacek, Internship<br>Implementation of Graph-Cut based Dense Multi-View Stereo Algorithm. |
| **2009/09 – 2009/12** | Micheal Vasiljevs, Internship<br>Image Processing with OpenCL. |
| **2010/06 – present** | Albert Kavlar, Internship<br>Edge-aware segmentation of images. |
| **2010/09 – present** | Franz Spitaler, Internship<br>Procedural Modeling of Facades |

## Peer-Reviewed Publications

[1] Przemyslaw Musialski, Christian Luksch, Michael Schwärzler, Matthias Buchetics, Stefan Maierhofer, and Werner Purgathofer. Interactive Multi-View Façade Image Editing. In Vision, Modeling, Visualisation (VMV'10), Siegen, Germany, 2010.

[2] Przemyslaw Musialski, Meinrad Recheis, Stefan Maierhofer, Peter Wonka, and Werner Purgathofer. Tiling of Ortho-Rectified Façade Images. In Spring Conference on Computer Graphics (SCCG'10), Budmerice, Slovak Republic, 2010.

[3] Matthias Baldauf and Przemyslaw Musialski. A Device-aware Spatial 3D Visualization Platform for Mobile Urban Exploration. In The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2010), Florence, Italy, 2010. IARIA.

[4] Przemyslaw Musialski, Peter Wonka, Meinrad Recheis, Stefan Maierhofer, and Werner Purgathofer. Symmetry-Based Façade Repair. In Vision, Modeling, Visualisation (VMV'09), Braunschweig, Germany, pages 3-10. DNB, 2009.

[5] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor Completion for Estimating Missing Values in Visual Data. In 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 2009. IEEE.

[6] Matthias Baldauf, Peter Fröhlich, and Przemyslaw Musialski. Integrating User-Generated Content and Pervasive Communications - WikiVienna: Community-Based City Reconstruction. IEEE Pervasive Computing, 7(4):58-61, October 2008.

[7] Matthias Baldauf, Peter Fröhlich, and Przemyslaw Musialski. A Lightweight 3D Visualization Approach for Mobile City Exploration. In First International Workshop on Trends in Pervasive and Ubiquitous Geotechnology and Geoinformation GIScience conference (TIPUGG'08), USA, 2008.

[8] Przemyslaw Musialski, Robert F Tobler, Stefan Maierhofer, and Charles A Wüthrich. Multiresolution Geometric Details on Subdivision Surfaces. In 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia - GRAPHITE'07, Perth, Australia, 2007. ACM Press.

[9] Przemyslaw Musialski, Robert F Tobler, and Stefan Maierhofer. Smooth Subdivision Surfaces over Multiple Meshes. In 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'07), Plzen, Czech Republic, 2007.

[10] Charles A Wüthrich, Jing Augusto, Sven Banisch, Gordon Wetzstein, Przemyslaw Musialski, Chrystoph Toll, and Tobias Hofmann. Real Time Simulation of Elastic Latex Hand Puppets. In 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'06), volume 13, Plzen, Czech Republic, February 2006.

[11] Andreas Emmerling, Kristian Hildebrand, Jörg Hoffmann, Przemyslaw Musialski, and Grit Thürmer. A System for Modelling in Three-Dimensional Discrete Space. In Discrete Geometry for Computer Imagery (DCGI 2003), pages 534-543. Springer, 2003.

## Other Publications

[12] Przemyslaw Musialski. Axis-Aligned Segmentation of Orthographic Façade Images. Technical report, VRVis Research Center, Vienna, Austria, 2010.

[13] Przemyslaw Musialski. Point Cloud to Model Registration. Technical report, VRVis Research Center, Vienna, Austria, 2009.

[14] Przemyslaw Musialski. Multiresolution Displacement Mapping on Subdivision Surfaces. Diploma thesis, Bauhaus-University Weimar, Germany, 2007.