Graz University of Technology

Institute for Computer Graphics and Machine Vision

Dissertation

Feature-Based Reconstruction of 3D Primitives from Multiple Views

Joachim Bauer

Graz, Austria, November 2009

Thesis supervisors

Prof. Dr. Franz Leberl

Prof. Dr. Horst Bischof

Für meinen Bruder Christoph (1974-2008)

Abstract

The modeling i.e. the capturing of geometric information plays a key role for our fast growing urban communities. Until recently the process of capturing three-dimensional data was a labor intensive task, but with the advent of high resolution digital imaging instruments a high degree of automation is within reach.

Urban architecture however, proves to be a very challenging territory for image-based modeling methods. Nevertheless modern photogrammetric and computer vision methods feature both, high robustness to cope with complex outdoor scenes and improved efficiency to allow the processing of huge amounts of data in reasonable time frames.

This work presents a collection of methods for the efficient feature-based 3D modeling of urban environments. High resolution digital images are the sole data source. The term feature-based modeling in this context means that the proposed methods do not directly work on the pixel-based image information, but higher level geometric features are extracted in an initial preprocessing step. Every subsequent method then operates on those primitives.

The topics presented span low-level feature extraction such as edges and ridges and corners via the robust detection of mid-level features such as ellipses and 2D line segments and new efficient methods for extracting 3D primitives from 2D features.

The main contributions of this work are methods for extracting vanishing points, robust fitting of regular polygons, a method for the efficient matching of points-of-interest via a semi-global descriptor and finally a method for the efficient feature-based 3D reconstruction from multiple images.

In the experimental section an in-depth analysis of the presented methods, concerning their robustness as well as their accuracy, is conducted. The experiments are conducted on synthetic as well as real data sets.

Keywords. Computer Vision, Photogrammetry, Feature-based 3D Modeling

Acknowledgments

A work like this is never a solo piece - many people and institutions contribute directly or indirectly to the success. I want to thank my former employer the VRVis Research Center and the partner company GeoData for their financial support during the first years of my research. I also gratefully acknowledge the support of the Institute for Computer Vision and Graphics at the Technical University Graz. Among the many co-workers and colleagues that went a part of the way with me, I want to give special thanks to Christopher Zach, Mario Sormann, Andreas Klaus, Lukas Zebedin and Konrad Karner. For many fruitful discussions and sheer endless patience during the course of the work, I thank Horst Bischof and Franz Leberl.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Background and motivation

A recent study of the United Nations Population Fund (UNFPA) shows that in 2007 about 3.3 billion people were living in towns and cities. By 2008 for the first time in history more than half of the world's population will be living in urban settlements or suburban areas that encompass only 380.000 square kilometers - that is less than the total area of Japan. The study further predicts that in 2030 more than 5 billion of the world's population will live in cities. Assuming 600 buildings per square kilometer in a high density area (city cores) and 300 buildings per square kilometer in suburban areas gives a rough estimate of 230 million buildings.

In order to keep track of such vast growing habitats a permanent monitoring concept is necessary. Given the complexities of larger cities and demands for increased levels of detail combined with the necessity for full automation brings traditional mapping techniques to their limits. Acknowledging this fact, it is only consequential to put efforts into the development of automatic systems that are capable of performing large parts of the mapping/modeling work flow. Newly emerging geo-information systems like Google Earth (http://earth.google.com) or Microsoft's Virtual Earth (http://bing.com/maps) support this argument. These systems provide efficient access to geo-referenced imagery, additional vector data (roads, borders, building-outlines etc.), meta-data (road names, names of places etc.) and recently also 3D content (mostly buildings and landmarks of significant interest). The great attraction that these systems provide can be explained by their intuitive user interfaces as well as their common availability that allows a person with access to the Internet to satisfy his/her urge to explore the world. These online map exploring systems are the answer to the problems that the average user experiences in dealing with complex environments such as large cities. Navigation is one of these challenges, be it either the directions to a user specified site or the closest route to the next business that provides a desired service.

These systems stand in sharp contrast to traditional city maps or maps in general, which are excellent examples for abstraction, since they provide only that amount of information necessary for navigation. The new web-based geographic information systems (GIS) are digital interactive maps that try to solve the abstraction problem by allowing the user to specify which information is displayed. Actually they provide an augmented version of the reality by combining real imagery with vector data and other meta data (real-time traffic information, weather).

At the beginning the birds-eye-view on the planet gives a new sense of freedom. This euphoria fades away when *real world* tasks like finding the location of the hotel for the next conference is on the to-do list. Up to now all imagery is mapped onto a '*bald earth*' model and the few existing building blocks are hand modeled and sometimes untextured. The lack of detailed elevation models makes the textures look flawed e.g. if tall buildings seem to lean away from the observer and even sophisticated blending methods can not conceal discrepancies at image borders. Figure 1.1 illustrates this effect for the downtown area of Houston, Texas: Due to the mapping of the aerial images onto the flat surface model the buildings seem to lean.

All these artifacts are unavoidable if no detailed building models are available. In order to overcome these shortcomings and to provide a more immersive experience for the user, fully textured and detailed 3D models of cities are necessary. The capturing of these habitats must be highly automated in order to provide a cost efficient alternative to traditional modeling methods. Buildings are the most prominent objects in an urban environment and therefore the recording of their geometrical as well as their visual

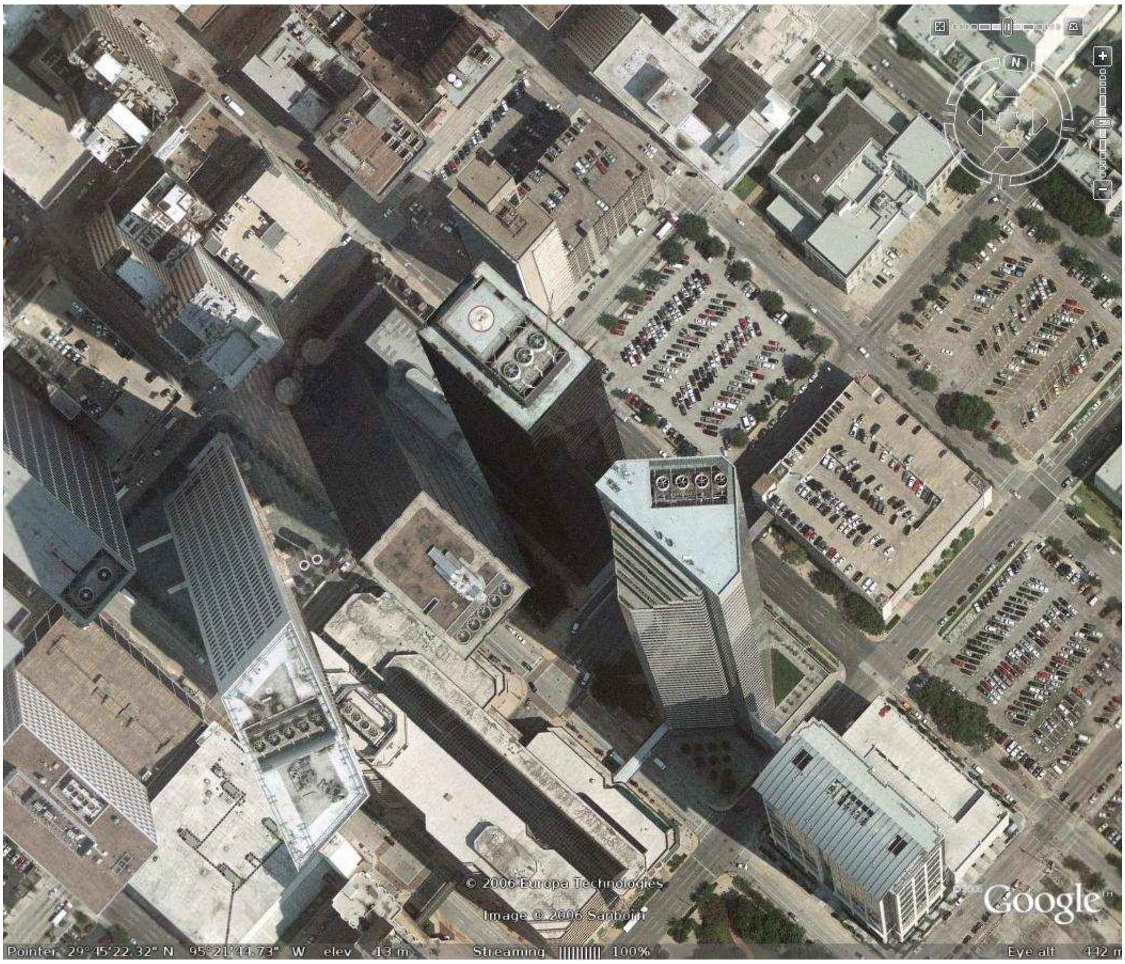

N
© 2006 Europa Technologies
Image © 2006 Sanborn
Google
Pointer 29°45'22.32" N 95°21'44.73" W elev 13 m
Streaming |||||||||| 100%
Eye alt 442 m

Figure 1.1: View of city center in Houston, Texas as provided in Google Earth. At the border of aerial images the skyscrapers seem to lean into different directions. This is caused by the traditional ortho-image generation method of mapping perspective images onto a 'bald earth' digital elevation model (DEM). Structures that deviate significantly from this DEM, suffer from perspective distortion.

appearance has been of increased interest since the early days of photogrammetry and even before the advent of photography, perspective drawings of cities were made e.g. the famous 'Huberplan' - a perspective drawing of the city of Vienna made in 1734. Therefore modeling and reconstruction of architectural objects has been a field of intensive ongoing studies.

3D city models can be roughly categorized according to one of the following five levels:

1. **Level 0** simple (polygonal) building outline + estimated height = block model, often modeled manually

2. **Level 1** block model + roof shape, modeled either manually or automatically using image or Lidar data

3. **Level 2** detailed $2\frac{1}{2}$D model + true ortho-image, modeled automatically from aerial images.

4. **Level 4** fully textured 3D model, modeled from aerial + street level images

5. **Level 5** abstracted 3D model with semantic information e.g. number of stories, windows, type of building etc.

1.1.1 History of city models

Man's need to measure his surroundings can be traced back to the ancient Egyptians and Greeks. The basic trigonometric principles developed more than 2000 years ago are still valid and build the foundations for today's applications. The measurement technologies however have changed dramatically since then. In the early 1400's artists began do amend their drawings by mimicking perspective cameras through the introduction of vanishing points. The foundations for photogrammetric measurements were laid in the mid 1800's, shortly after the invention of photography by Joseph Niepce in 1834. The first use of photographs for the extraction of geometric information was probably the work of the French officer Aime Laussedat (1819-1907) who used terrestrial images in 1851 (and later on aerial imagery acquired from kites) for creating topographical maps. While this technique originally was called 'iconometry' he is nowadays referred to as the 'Father of Photogrammetry'. The first reported photogrammetric measurements of buildings date back to 1858 and were performed by the German Albrecht Meydenbauer (1834-1921). He developed the predecessor of modern metric cameras in 1867 and cued the term ‘Photogrammetry’ [47]. In 1866 the Austrian physicist Ernst Mach published his idea to use the stereoscope to perform volumetric measures. Carl Pulfrich presented the first stereo comparator in 1902 and revolutionized the process of mapping from stereo pairs. In 1921 Sherman Fairchild produced an aerial map of Manhattan Island composed from one hundred overlapping images. During the 1930s the bundle block adjustment methods were developed based on Carl Friedrich Gauß's calculus of observations.

Aerial images have played an important role in city modeling since the early days of photography. Accurate mapping was done by geometric analysis of images. The cost of analogue film and the labor intensive process of performing the estimation of the exterior camera orientation parameters, lead to an efficient and well defined work flow. As a consequence, as few images as possible were taken during a mapping mission in order to reduce the significant amount of manual work involved in the determination of the exterior orientation based on measuring ground control points and especially in the production of elevation models, where corresponding points are measured in a stereo pair. Due to this high amount of manual interaction the resulting elevation models typically exhibit a relatively large grid size (tens of meters).

The modeling of urban environments was and often still is restricted to the survey of isolated buildings or single building blocks. So up to now the task of modeling larger portions of a city is a time consuming task which involves a lot of manual work. The creation of a 3D city model with a little more geometric detail than simple building blocks composed from planar facades requires a lot of manual interaction. Due to these limitations the resulting models are bound to a high level of abstraction. However for many applications these coarse models are sufficient.

1.1.2 Modern mapping systems

With new sensor technologies emerging and continually growing processing power, the creation of large scale city models is now becoming a feasible task. The availability of lower cost digital cameras and fast laser scanners has led to a new boom in 3D city scanning/mapping. Each mapping project aims at different goals and these goals are used to define individual specifications for the capturing missions. Based on the sensors used, the generated model varies in the level of detail as well as in the geometric primitives that result from the modeling process ($2\frac{1}{2}$D mesh, point cloud, 3D primitives). State-of-the-art methods for modeling record 3D information either directly e.g. range finding devices (total station, 3D laser scanner) or indirectly via photogrammetric methods. Based on the sensing technology the distinction between mapping systems can be drawn. However the largest differences manifest themselves with a closer look at the mapping platform. The three main classes of mapping platforms are space borne systems, airborne systems and terrestrial systems. Space borne platforms are satellites that perform large scale or global mapping tasks and use highly specific sensors like synthetic aperture Radar, multi spectral cameras et cetera. The class of airborne platforms encompasses airplanes, helicopters and more exotic platforms such as balloons, airships, kites et cetera. Terrestrial platforms can be all forms of motor vehicles, but also much simpler ones such as a camera or a laser scanner on a tripod. Also wearable systems become more popular with the shrinking size of sensors. The choice for a specific platform depends on the project area and on the type of data that is to be recorded. For standard mapping applications where larger projects can span entire districts, airborne platforms are the method of choice. Vehicle-based terrestrial platforms play a role in the creation of large scale street level city maps or mapping the condition of roads. For the precise mapping of individual sites hand held cameras or tripod-mounted laser scanners are used. In the following a short taxonomy of sensor systems will be given.

One reason for this is that in contrast to aerial mapping the work flow for terrestrial mapping is not so well defined.

Passive sensors Passive 3D measurement devices are restricted to process the information that the scene provides - which is mainly visual information. Theodolites are high precision instruments for angular measurements but are nowadays more and more replaced by total stations (see 1.1.2). The most common passive 3D sensors are cameras. For the image-based acquisition of geometric information numerous methods exist. The earliest were developed in the 19^{th} century for the purpose of photogrammetric surveying. Modern methods that derive 3D data from images are generally referred to as 'Computer Vision' techniques. All those image-based 3D measurement techniques rely on the triangulation principle.

The use of modern vision algorithms theoretically allows for high precision image-based measurements but image-based modeling methods have the disadvantage that the accuracy of the achievable 3D measurements depends strongly on the geometric configuration of the camera setup. This fact is acknowledged by using simultaneously triggered stereo or multi camera setups, where the relative pose (rotation and translation) between the cameras is known and consequently 3D data can be extracted for each 'shot' [124].

Active sensors Active sensors emit well defined signals that allow for a robust measurement of distances from the instrument. The signal can be a pulse of light (laser) or sound (especially ultra sound) or electromagnetic radiation (e.g. Radar). For highly accurate measurements the distance measurements are carried out with laser-based instruments. Total stations are measuring devices that are used for high precision measurements at a low measurement rate (typically several points per minute). They are used at construction sites as well as for architectural documentation. For surveying larger parts of a city this type of instruments is inapplicable.

Another class of direct 3D sensing instruments are 3D laser scanners which are capable of generating vast amounts of reasonably accurate 3D measurements. The measurement principle is based on time-of-flight measurements for the emitted laser pulse. A subclass of these instruments are profile scanners which acquire depth measurements in one dimension only. In order to capture depth information in two dimensions these scanners are typically mounted on moving platforms. A good example are the airborne Lidar platforms where a profile scanner measures reliable depth values, while an accurate GPS/INS unit provides position and bearing information of the platform (slow flying planes or helicopters). This sensor combination has brought new impetus to the field of automatic DEM generation, since the error prone generation of dense 3D data is solved. The filtering and interpretation of Lidar data is now the main course of research.

An improved version of the profile scanners are the 'panoramic' 3D scanners that are capable of capturing depth information across most of the hemisphere but need a significant amount of time for completing a full scan. This is due to the fact that the instrument can only measure one point after the other by rotating a reflecting mirror about the two spherical axis. The result of the scanning process is a depth matrix that can be interpreted as $2\frac{1}{2}$D height-field.

The advantage of direct 3D measuring devices is the reliability of the recorded data. The average measurement error is known in advance and independently of the working distance. On the other hand the recording rates of total stations is several points per minute and for 3D laser scanners like the Riegl LMS Z360 [101] it is 11.000 points per second. During recording the device must not move and it takes in the order of tens of seconds to minutes to complete a scan at finer resolution. Thus for recording the geometry of buildings from street level the scanner platform operates in a stop-and-go mode moving from one capturing position to the next. For large scale city modeling where several kilometers of facades have to be captured such a strategy is not sufficient.

A somewhat more efficient approach was demonstrated in [41], where two profile laser range scanners are used in an orthogonal setup. One scanner measures vertical profiles and the other one scans in the horizontal plane. The profile scanners record several tens of profiles per second and thus allow for a mobile platform. However, due to the moving platform the relative orientation between consecutive 3D profiles is unknown and a registration step has to be performed in order to transform all measurements to a common coordinate system. The orthogonal arrangement of the profile scanners is used to extract the rotation and the translation parameters of the scanning platform. Assuming a rigid scene consecutive horizontal profiles are registered by a variant of the iterative closest point (ICP) algorithm. The vertical profiles are aligned by detecting linear structures in the profile and the assumption that these structures belong to dominant vertical facade planes. The registered 3D profiles are then transformed into a common coordinate system, converted into a mesh representation and finally textured from images of a camera that are calibrated to the profile scanners. This computationally highly complex approach finds its counterpoint in commercial applications such as StreetMapper [48]. For this system the task of geo-registration is performed from the data of a high performance GPS/inertial navigation system. In general, modern mapping systems are realized by a combination of different sensors. These additional sensors either augment the captured data by recording data that can not be sensed by the main instrument or they are even an essential component whose data is crucial for the processing. In the following a short survey of such augmenting sensors is given.

1.1.2.1 Augmenting sensors

The combination of several sensors on a capturing platform provides enriched information but results also in the problem of sensor fusion and registration of data from different sensors. Examples are electronic compasses, GPS receivers or inertial sensors (IMU's); These sensors alleviate the task of computing the exterior orientation of imaging sensors by producing good estimates for the platform orientation and position. Especially for airborne platforms the combination of a GPS receiver with a high precision inertial sensor is often used to provide accurate estimates for the image orientation.

Whereas GPS receivers provide accurate position information when used in an aerial mapping system, they suffer from a substantial loss of accuracy when used in a mobile terrestrial setup. For example the position coordinates are corrupted with noise due to multiple reflections of the satellite's signal on facades. When driving through very narrow alleys or tunnels the GPS sensor might not receive any signal at all and the only way to provide position and orientation estimates is the use of an IMU sensor.

Another frequently used sensor in terrestrial mapping systems is a wheel encoder that provides information about the vehicle's velocity, or in the case of multiple sensors on different wheels, also information about the heading of the mobile platform. As mentioned before, the main problem for the combined sensor platform is fusion of data coming from individual instruments. Every sensor has its own capturing rate which may vary from several tens of readings per second for the odometer to a couple of GPS readings per second to less than one image per second captured by the cameras. The same inhomogeneity holds for the accuracy provided by the individual sensors.

1.1.3 Mapping scenarios

In order to capture large areas in an efficient manner the choice between two dominant methods, namely photogrammetric methods and laser scanning, has to be made. For airborne platforms photogrammetric methods are preferred, but lately laser scanning also showed promising results. For terrestrial mobile platforms laser scanning dominates since the on-site data recording times are short and the post-processing can be automated to a high degree.

1.1.4 Aerial mapping

Airborne mapping has traditionally been performed with cameras. The process of image acquisition, image orientation, geo-referencing and 3D data generation is very well structured and documented in numerous publications. The range of products that are generated encompass digital surface models (DSM's), digital terrain models (DTM's), ortho-images, road networks et cetera. Images are acquired by flying a pattern of parallel lines over the project area - this results in strips of overlapping images.

Traditionally the 3D data acquisition for creating maps was accomplished by manual measurements taken from aerial photographs. This strategy made perfect sense as long as only prominent features were measured e.g. roads, larger buildings or isolated landmarks. As soon as the number of measurements increases the manual picking of points becomes a tedious task. Nevertheless manually assisted modeling methods are still state-of-the art [1]. The advantage of this strategy is that humans are good at abstracting complex geometry. Typically a model is then built in a hierarchical manner by performing a natural coarse-to-fine modeling strategy. This is especially useful when the resulting models should be as simple as possible (in terms of facets per building model) in order to allow a fast transfer over networks.

With the advent of digital aerial cameras the degree of overlap between the images has drastically increased [74]. The increased redundancy allows for the robust automation of the workflow.

1.1.4.1 Terrestrial mapping (Streetside)

Building modeling from terrestrial photographs differs from the aerial modeling path in several ways. The camera viewpoints generally lie on a complex path and automatic orientation becomes significantly more challenging. The camera path is constricted by narrow streets and it is often a non-trivial task to have every surface patch of a facade covered by a sufficient number of images (at least two for stereo). In contrast to aerial-based surface models which traditionally are $2\frac{1}{2}$D, a city model is 3D. This fact makes modeling of large areas especially challenging.

1.1.5 Applications

In the following subsection an overview of the various applications of of spatial information will be given. Furthermore the connection to the 3D data that are generated by the methods developed in this project are explained.

1.1.5.1 City maps

3D city models are met with a growing demand. Various business models depend strongly on the existence of spatial data. In the following a short survey of the most prominent applications will be given.

Recent web-based applications by global providers like Google maps (http://maps.google.com/), Microsoft's Bing maps (http://www.bing.com/maps/) as well as solutions by local providers such as Herold (http://www.herold.at/routenplaner/) or Klicktel (http://www.klicktel.de/kartensuche/) show the potential of modern geographic information systems. They provide easy access to map data and are continually working on enriching the contents (business locations, road vector data, content provided by the web community etc.).

A better term for these types of applications would be 'geographic entertainment' systems, since many users consider these web-based platforms as a social meeting place. Especially community based interaction shows that the information system acts as a platform for sharing localized information.

These systems initially were filled with digital vector data and subsequently augmented by ortho-imagery from satellite cameras or from national aerial ortho-photo programs. Trends show that in the future the main focus will lie on capturing and modeling information at the 'human scale', thus focussing on the 'urban canyons'. Those areas, which consist mainly of vertical facades are hard to capture from aerial views - roof overhangs and dense vegetation obstruct the clear view of facades. Mobile image acquisition platforms will be used for the efficient mapping of the major routes within a city. Automated street side mapping began in the mid 1990's. A first version of such a system was presented by Google [46] - it allows the seamless navigation through the captured streets and offers a panoramic 360 degree view. For navigation tasks it provides a good visual impression from a set of predefined viewpoints. However, since the image data is unedited it represents a snapshot taken at a certain date and time. Due to the fact that no 3D information is present the navigation path is restricted to lie on the path of the capturing vehicle. In fact the system is basically a collection of geo-referenced images that can be transformed into a seamless stream.

A real 3D model however supports the visualization from arbitrary view points, allows to take measurements and permits the insertion of artificial objects in order to augment the experience. A good example for the use augmentation are car navigation systems where the route is dynamically overlaid on the model. The realization of such a model constitutes a significantly greater challenge to the creators. The only feasible way to derive 3D models is a fully or nearly fully automated workflow. However, in order do derive a 3D model in an automated way the collection of input images must provide sufficient redundancy (overlap between images). Many transient objects, which are mainly vehicles and people, can not be modeled and must therefore be detected and ignored in the texture generation process. Furthermore, complex objects such as trees and bushes are hard to model. Thus, a semantic interpretation framework seems to be essential. Such a framework would detect generic object classes in the images and drive the 3D modeling.

1.1.5.2 Planning / Real Estate

For planning in densely packed urban areas a detailed 3D model helps to assess the visual impact of planned constructions. City officials, architects and contractors are able to easily grasp the situation on site. Public participation is made easier for people who are not skilled in reading 2D plans. High quality renderings give a realistic impression of the project before the first brick is laid. Another driving force is the multi-million real estate business. Aerial views allow potential customers to assess the neighborhood and infrastructure around an estate in a very efficient manner.

1.1.5.3 Tourism / Cultural Heritage

With so many places to visit in a strange town, tourists can use a visually appealing model to plan their visits to important sites and also for faster acquiring a feeling for navigating in an unknown town. Hotels and restaurants can place advertisements and cultural events can be announced and be pinpointed to.

Archeology is a field which suffers from a chronic lack of money. A cost efficient web-based presentation of excavation sites to a broad audience could help raising funds and at the same time serve as an education platform and help to popularize the results of archeological research. Previous projects like 3D Murale [90] showed a high potential in creating 3D models of those sites but failed in creating an ongoing interest due to a lack of a standardized web-platform for serving the content to the interested community.

1.1.5.4 Communications

Wireless communication can be optimized by analyzing the signal propagation and placing a transceiver station on the ideal location. While for wave propagation simulations a coarse building outline may be sufficient, the physical properties of the facades may be of interest. This application became very relevant in the late 1990's, when the mobile telecommunication boom started.

1.1.5.5 Defense / Public safety

Urban-wide hazard analysis benefits from detailed 3D models which allow e.g. for an accurate simulation of the spreading of harmful gases. Industries that deal with hazardous products can be placed on locations where they are least harmful in the case of an accident. Fire fighters and counter-terrorism forces can both make use of detailed facade plans that provide information about window placement and other ways of getting access to a building. Assessment and rescue planning becomes important. For those types of applications an abstracted visualization might serve better than a photo-realistic rendering and robust methods for automatic recognition and measurement of facade elements (e.g. number of stories, location of windows, doors etc.) become important.

1.1.5.6 Gaming

Interactive games that are set in realistic looking environments that represent real world locations give the player the feeling of better immersion. Photo-realistic models with a low polygon count are desired in this context.

1.2 State-of-the-art

The 3D modeling of buildings from images is a topic of ongoing research since the early days of photogrammetry. What is new is the need for, and interest in achieving full automation. The two main areas of research are modeling from aerial sensed data and modeling from terrestrial data. The creation of models from images was traditionally a field for photogrammetrists and the goal was to perform accurate measurements of geometric features. Typically a small number of points was measured and converted into line drawings which serve as a good abstraction of detailed facades. Another typical product is the so called ortho-photo of a facade - this is basically a perspective image transformed into a parallel projection. Generally the photogrammetry work flow for generating 3D information from images involves a significant amount of manual work.

Since the computer vision community has adopted this field the degree of automation has been drastically increased. The range of products that are derived from input images has also increased and often the geometric accuracy has to give way to visually appealing models.

The methods used differ in amount of manual input, types of sensors used and resulting models. The two main data sources are digital images and laser sensed range data (Lidar) - in this project we want to concentrate solely on image data. While the airborne methods are mainly concerned with map building and classification, the terrestrial methods are traditionally aimed at 3D modeling e.g. for architectural documentation. The main difference between computer vision and photogrammetry is, that the latter is a standardized technology and industry and less of a research area. In general photogrammetry approaches define a complete work flow for a specific problem (e.g. from digital images to photo realistic models) while in computer vision often only specific problems are tackled (e.g. wide baseline matching, camera pose estimation, reconstruction). An overview of existing city modeling techniques can be found in [55].

1.2.1 3D modeling from aerial images

Aerial-based modeling approaches are used to generate 3D models of large urban areas. Commercial products for large scale modeling, like CyberCity modeler [1, 129] are mainly using aerial images to compute ortho-photos, digital elevation models (DEM's) and 3D building models. Geo Systems [120] offers a product for the generation of detailed 3D models that are created with their semiautomatic modeling approach. The degree of automation in the early processing steps of aerial sensed data is high. This is achieved mainly through the use of additional sensors such as high accuracy GPS receivers and inertial navigation systems (INS). These sensors provide an accurate estimate for the camera pose and therefore allow for a robust triangulation. Methods for automatic generation of digital surface models (DSM) and the derived digital elevation models (DEM) do exist, however the reliable automatic detection and modeling of building primitives is still a topic of ongoing research [67].

Research projects aim at fully automated systems. In [62] a system for automatic detection and reconstruction of buildings from aerial images is presented. The output of such systems ranges from camera poses to photo-realistic textured 3D models. The automatic land-use classification problem however is still a field of ongoing research. Other topics are the automatic extraction of road networks [15], detection and classification of buildings and vegetation. All these methods aim at the fully automated extraction of data that can be fed into the various abstraction layers of a GIS system. The emerging of new digital high-resolution sensor systems such as the UltraCam D [75] brings new impetus to the field. These new airborne sensors are capable of capturing large urban areas with a significantly higher overlap of images than was possible with film-based aerial cameras. A typical overlap ratio is 80% in flight direction and 60% across neighboring strips. This increased redundancy allows for processing of data with a much higher degree of automation and increased robustness.

1.2.2 3D modeling from terrestrial images

In contrast to aerial-based modeling methods, the automated terrestrial modeling still faces a number of challenges. The scenes are more often complex because man-made structures represent ambiguities, occlusions exist, translucent and specular surfaces make an automatic interpretation hard. Semiautomatic approaches are preferred, since automated approached are yet to evolve. Current commercial solutions are [58, 100].

Modern 3D computer vision provides powerful methods for the automation of the work flow's main stages. Despite intensive research in the field, the main hurdles in the terrestrial 3D modeling work flow for street side images are the establishment of robust point correspondences between the images and the generation of a consistent dense 3D model. The correspondence problem has been recognized to be a hard problem and countless publications report on solutions. Recently there has been substantial progress in this area using invariant local features and probabilistic modeling [61, 78, 83–85]. These local approaches have demonstrated considerable success in a variety of applications, like recognition of objects [22, 39, 73, 105], wide-base line stereo [126], robot navigation [37, 44, 114], and image retrieval [63, 133].

The general task of generating 3D models from images has been one of the central goals in computer vision, applicable to areas such as reverse engineering, cultural heritage, building reconstruction, etc. [62, 116]. The structure and motion problem (recovering the

3D scene structure and the camera motion simultaneously) [7, 68, 79, 95, 96, 118] has already reached a state of maturity.

However, in the area of street side modeling one has to cope with repeating patterns, significant changes in view point and camera orientation (so called wide-baseline setups), occlusions, variations in illumination, highly reflective or translucent materials and last but not least dynamically changing scenes. Recent developments that are based on the extraction of affine covariant features increased the robustness of image correspondence estimation and the subsequent camera pose estimation.

On the other hand, man-made objects exhibit properties that alleviate the detection of corresponding points. Building facades are often planar structures and contain straight line segments. A special property of building facades is the frequent presence of groups of parallel line segments that form strongly distinct vanishing points in images. The knowledge of two vanishing points that belong to orthogonal sets of 3D lines allow a robust estimation of the relative rotation between successive camera view points. Vanishing point detection techniques use Hough-like voting schemes [103, 128], voting methods on the unit sphere [27] or grouping strategies [108].

The detection of planar structures via the use of homographies is another frequently applied method [6, 97, 104, 134, 135]. Homographies can be extracted from four point correspondences between an image pair and allow a direct point-to-point transfer, thus imposing stronger constraints on the correspondences than an arbitrary relative orientation described by the fundamental matrix. The presence of rectangular structures such as windows or doors has been exploited in [98] to extract local homographies that are derived from those parallelograms. Ellipses or elliptic arcs are another class of primitives frequently found in architectural images. Robust detection methods for ellipses or conics are described in [25, 94, 99].

The automatic detection of vanishing points, homographies and ellipses are examples where a specific geometric property of buildings is used to improve image matching or object modeling methods.

Repeating structures, such as identical windows, friezes or stucco work, are problematic when correspondences are computed by matching local descriptors alone. However, recent publications show that a combination of local and global descriptors can overcome some of the matching ambiguities. The method proposed by Mortensen [89] combines global features that were initially used to compare 2D shapes with local descriptors in order to provide a more robust similarity measure. Tell and Carlsson [123] proposed a method that incorporates the topology of neighboring points via intensity profiles to improve the stability of the matching process.

After establishing point correspondences between image pairs, the parameters of the relative orientation between the two images can be computed. Finally the pairwise image correspondences are linked over multiple images and the exterior orientation parameters of an image sequence are computed using bundle adjustment [76]. The methods mentioned above result in a sequence (a 'block') of oriented images and sparse 3D primitives (points, lines). In order to bridge the gap between sparse reconstructions and the desired surface model a dense stereo or multi-image matching technique is applied. These depth estimation methods assign a distance value to every pixel in a reference image. Numerous approaches to dense stereo matching have been published [19, 30, 52, 119, 137]. Those methods are global approaches that are determining all disparities simultaneously by applying energy minimization techniques such as graph cuts, belief propagation, dynamic programming, scan-line optimization or simulated annealing. The accuracy of a dens matching result is being defined as the error in depth (distance from the camera). This error is a function of the intersection geometry of the optical rays, of the matching accuracy (influenced by texture similarities, specular surfaces etc.), and of differences in the overlapping images.

Recently, segment-based methods [19] have attracted attention due to their convincing performance. They are based on the assumption that the scene structure can be approximated by a set of non-overlapping planes in the disparity space and that each plane is coincident with at least one homogeneous color segment in the reference image. Especially for the modeling of buildings, the segment-based approaches seem to be promising.

When it comes to the modeling of larger portions of a city from ground views, few practicable approaches have been published. The research group lead by Seth Teller has published several papers [2, 3, 117, 124] concerned with the creation of detailed building models from digital street side photographs.

Fully automatic approaches such as [41] combine active sensors such as laser range finders with digital imaging sensors for robust recording of 3D structure and texture.

A problem that still draws the attention of researchers is automatic abstraction of the dense surface data that are generated in the modeling process. The need for fast transmission over limited bandwidth connections and the huge amounts of geometrical as well as texture data that are generated with automatic modeling methods make a data reduction step that is usually achieved by abstraction necessary.

With approaches that work on the geometric properties of the model itself [43, 53], the degree of abstraction versus the visual appearance is limited. Systems that automatically recognize recurring structures or dominant portions that can be modeled with simple analytic surfaces (e.g. planes, cylinders or parametric surfaces) have the potential to reach significantly higher amounts of data reduction, while still retaining a good visual appearance.

1.3 State-of-the-art of 3D modeling

Many ways exist to traverse the path from input images to a 3D model, but generally 3D modeling methods branch into two large groups: image-based modeling [19, 30, 52, 119, 137] and feature-based modeling [64, 112, 135]. While the field of image-based modeling is extremely popular among researchers, the number of publications that deal with feature-based modeling is much smaller. The main reason for this imbalance is the fact that dense surface models are perfectly suited for visualization and can easily be interpreted/understood - even by an untrained user. The result of feature-based modeling approaches is often a sparse cloud of points or lines and typically lacks many of the details present in dense 3D models. Thus, sparse point clouds constitute a significantly harder challenge for an interpretation. Most image-based modeling methods use area-based matching algorithms which are known to be problematic at sharp edges or depth discontinuities.

3D primitives that are derived from features often contain complementary information that can not or only with high effort be extracted by image-based modeling. This fact makes feature-based modeling methods valuable, despite the fact that they can not be used as a stand-alone 3D reconstruction approach. They need to be seen as a supplement to enrich the overall quality of the derived model.

Historically, feature-based modeling was the only way to extract geometric information from a set of images by hand. The labor intensive collection of manual image measurements allowed only for creating very sparse sets of 3D primitives.

Many manual photogrammetric applications still provide only means for measuring 3D points and 3D lines. An important reason for this restriction is the fact that these simple 3D primitives can be derived from image measurements using well defined geometric algorithms. Especially in the case of multi image vision this property is important. The class of methods that estimate the optimal 3D primitive from 2D measurements in a set of images are the so called EM algorithms [38] pp. 357 ff. For problems related to the estimation of geometric 3D primitives, the criterion that is minimized is the squared reprojection error i.e. the sum of all squared distances of the 2D image measurements with respect to the projection of the reconstructed primitive. Another important property of simple 3D primitives is that the achieved accuracy of the reconstruction can be evaluated by means of error propagation [122] pp. 260 ff. This fact, together with the above mentioned capability of providing accurate geometric measurements on depth discontinuities, makes feature-based modeling approaches especially interesting.

Dense area-based reconstruction methods (as in contrast to feature-based) methods are bound to minimize an energy function. The energy function typically tries to find a good trade off between the smoothness of the reconstructed surface and the data fidelity. Recent approaches use energy minimization schemes like total-variation optimization [87] or graph cuts [69] to extract the 3D surface. Due to the fact that the optimization criterion does not directly measure the reprojection error of the created 3D surface points these methods can not provide measures for geometric uncertainty of the resulting surface. Another drawback of dense multi view modeling methods is the complex handling of visibility - a 3D surface evolves during the optimization process, which creates dynamically changing visibility conditions that pose great challenges to true multi-image modeling methods. Many dense surface modeling methods circumvent this by creating 3D surfaces for each image pair by means of dense stereo matching methods and obtain the final model via a subsequent fusion step [138].

Geometric image features give very strong visual cues and allow a good perception of the prominent characteristics of a scene from a small amount of data. Figure 1.2 illustrates of this effect: despite the fact that only strong edge features are present the prominent landmark (clock tower in Graz) can be clearly recognized. These contour segments capture the most important shape properties of the object. Thus, a line drawing is nearly as descriptive as the image itself, but needs significantly less memory to store it. The challenge is to convert the set of features from multiple images that show the same object into a meaningful 3D description. Several publications have shown that this is possible [64, 112, 135].

1.4 The project

Our interest lies in automated methods for creating 3D city models from overlapping images. The focus lies on the extraction on 3D modeling based on geometric image features. In the course of this project methods for the extraction of geometric 2D features from single images will be presented. It will be shown how these geometric features can be

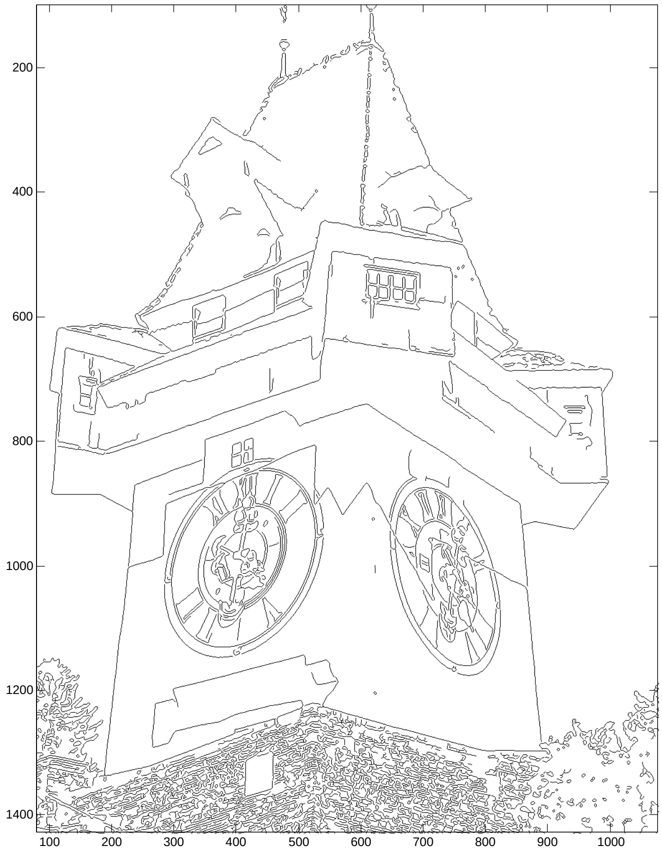


Figure 1.2: Canny edges for a well known landmark in Graz - can you recognize it? - it is the famous clock tower (Uhrturm).

used to establish point correspondences between pairs of images and finally 3D modeling approaches based on geometric image features will be introduced and discussed. The goal is to show that 3D structure from geometric image features can be derived with high geometric accuracy. The source data will be high resolution digital images which show objects with a great redundancy. The fact that an object point is visible in several images improves the robustness of the 3D modeling approaches.

1.5 Outline

The rest of the document is structured as follows:

1. **Feature extraction** We first introduce various classes of geometric image features. Beginning with point-like features (edges, ridges, corners) we advance to features based on grouped points (contours, straight line segments). We present methods for the robust detection of vanishing points from sets of straight line segments. The last two sections of the chapter are on features that are computed by estimating geometric primitives from point-like features (2D line segments, ellipses, affine transformed squares). Experiments are performed to assess the geometric accuracy of the extracted features.

2. **Feature matching** We then proceed to the establishment of correspondences between features and introduce the various types of so called transform invariant image descriptors. We introduce state-of-the-art descriptors and a new semi-global, rotation-invariant descriptor.

3. **3D modeling** Finally we extract 3D primitives from sequences of oriented images. Methods for detecting 3D points, 3D lines and 3D planes are presented. This material presents the main contribution of the project, namely a method for efficient 3D reconstruction of object contours. In the experimental section we finally show that the proposed method is applicable to a wide range of applications and that the achievable accuracy is sufficient to serve as valuable input for subsequent modeling approaches.

4. **Discussion based on data** The performance of the presented methods for synthetic as well as real data is shown. The setup of the experiments is explained and the results are discussed.

5. **Conclusion and outlook** The final chapter recapitulates the main contributions, discusses the findings and closes with an outlook on future work.

1.6 Key Publications

The main publications that are the basis for the presented work are reported in the following list. The aims and the key findings of each paper are shortly described.

- Bauer, J. and Klaus, A. and Karner, K. and Zach, C. and Schindler, K., METROPOGIS: A Feature Based City Modeling System, *In Proceedings of the ISPRS Comission III Symposium, Graz, 2002, pp. 22 - 27*, [11]; In this publication, a feature-based frame work for generating 3D primitives from multiple oriented images of building facades was presented.

- J. Bauer and H. Bischof and A. Klaus and K. Karner, Robust And Fully Automated Image Registration Using Invariant Features, *In Proceedings of ISPRS - Int. Society for Photogrammetry and Remote Sensing, 2004, pp. 12 - 23*, [10], This paper reports on a method for the robust extraction and description of points-of-interest by intersection straight line segments. The generated primitives, called 'Zwickels', are used to establish robust correspondences between image pairs.

- J. Bauer, K. Karner, and K. Schindler, Plane parameter estimation by edge set matching, *In Proceedings of the 26th Workshop of the Austrian Association for Pattern Recognition, 2002, pp. 29 - 36*, [60]; This article describes a method for the efficient generation of 3D plane hypotheses by performing a feature-based sweep approach.

- J. Bauer, A. Klaus, M. Sormann, K. Karner, Sparse 3D Reconstruction by Edgel Sweeping, *In Proceedings of the CVWW - Computer Vision Winter Workshop, 2004, pp. 11 - 20*, [59] This is the first of three publications on the feature-based space sweeping approach. The generation of 3D primitives from directed 2D image features (edgels) is demonstrated.

- Bauer, J and Klaus, A and Sormann, M and and Karner, K, Efficient 3D Reconstruction by Edgel Sweeping, *In Proceedings of Optical3D (Optical 3-D Measurement Techniques), 2005, pp. 253 - 262*, [12] This is the second publication on the feature-based space sweeping approach. Improvements in the robustness are presented.

- Joachim Bauer and Christopher Zach and Horst Bischof, Efficient Sparse 3D Reconstruction by Space Sweeping, *In Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission, 2006, pp. 527 - 534*, [13] In this work an improved sweeping approach, based on rectified images is presented.

Chapter 2

Feature extraction

Contents

2.1 Geometric image features

The extraction of geometric image features is one of the most important tasks of a modern computer vision system, it not only reduces the amount of data to be processed, but also increases the robustness and accuracy of measurements in digital images. Geometric image features are primitives that are created by performing transforms on the raw image data. Most image features can be assigned to one of the following categories:

- Point-like features such as edges, ridges and corner points. These are pixel-level features i.e. they can be calculated at each pixel using e.g. derivatives or color information at each pixel.

- Region-like features as e.g. watersheds, MSER's,

- Higher geometric primitives arising from grouping operations such as contour lines, or features detected by fitting geometric models to sets of point features like straight line segments or conics (ellipses, parabolas, hyperbolas).

Another categorization can be made with respect to the application the features are used for: For establishing image matches, point-like primitives with a high spatial accuracy and good repeatability under various geometric transforms are preferred, while for image segmentation region-like features play an important role. In the field of 3D reconstruction point features and higher geometric primitives such as straight line segments or conics are of special interest. The reason is that these features can be simultaneously used to compute the camera's pose as well as the 3D scene structure. Furthermore a theoretically sound model for estimating the optimal 3D primitive as well as its uncertainty is available [9, 51].

The classification given above is not unique, but we define a hierarchy on the feature classes that will be used within this document. Low-level features are extracted directly from the original images, whereas higher level features result from geometric operations performed on low-level features.

2.1.1 Point-like features

The most basic feature is a point. Depending on the extraction method additional scalar attributes can be associated with the feature point. Corners for example are locations in the image where both eigenvalues of the 2×2 structure tensor attain large values. The strength of a corner is a function of the structure tensors determinant and trace and is encoded as additional scalar. Corner locations are very distinct in an image and thus corners play an important role in image matching methods and the means for their extraction are described in several publications [40, 49]. Features that are found by searching an image's scale space e.g. by detecting local maxima of difference-of-Gaussian filter results as proposed by Lowe [77] have their principal scale as additional attribute.

The next class are oriented point features. The two main types are edges and ridges. An edge element (short *edgel*) describes an image location where a strong intensity difference to one of the neighboring pixels occurs. Edgels can be extracted very efficiently using the method of Canny [24]. Due to the use of differential geometry operators not only the location of the edge but also the direction of the gradient is determined. Many higher level geometric primitives are formed by grouping edgels, so these simple features form the basis for many subsequent algorithms. Ridges are another class of oriented point features. Ridges are used to detect line-like structures in the image. The extraction of ridges is also performed by methods of differential geometry, in this case by analyzing the Hessian matrix. The orientation vector associated with ridges points either in direction of the ridge (= minimal curvature) or perpendicular to it (= maximal curvature).

2.1.2 Region-like features

The class of region-like features plays an important role in segmentation applications. Examples for region features are watersheds, maximally stable extremal regions (MSERs) proposed by Matas et al. [61] or regions defined by clustering methods and subsequent grouping e.g. mean-shift regions proposed by Comaniciu and Meer [28]. Region-like features will not be covered in this document.

2.1.3 Higher level primitives

All the afore mentioned features can be used to build higher level primitives. This can be achieved either by grouping, where points are connected to form chains or by fitting a geometric model to a set of points. The most prominent features are straight line segments but also the family of conics (parabolas, hyperbolas and ellipses). Rather exotic features are the recently published affine superellipses [92, 136].

The following sections provides more details on the various feature extraction methods, limiting ourselves, however, to the most important ones. The structure is hierarchical and starts with a short introduction to edge and ridge extraction. Next is the extraction of corners. One level above the simple point-like features are the geometric primitives that are generated by grouping those point-like features, namely contours and straight line segments. A special interest exists in the fitting of primitives to sets of point-like features. Finally we discuss two methods for extracting vanishing points.

For denoting image operations we follow the notation of Lindeberg [121] and define: An image f, and a Gaussian kernel g:

$$g(x;\sigma^2) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \tag{2.1}$$

where σ is the standard deviation of the Gaussian kernel. The convolution of the image with the Gaussian kernel is

$$L(:;\sigma^2) = g(:;\sigma^2) \star f. \tag{2.2}$$

The derivatives are then defined by:

$$L_{x^\alpha y^\beta}(.;\sigma^2) = \partial_{x^\alpha y^\beta} L(.;\sigma^2) = g_{x^\alpha y^\beta}(.;\sigma^2) \star f; \tag{2.3}$$

Thus the convolution of the image with the first-order derivative of Gaussian kernel with respect to x is denoted as: L_x and the convolution of the image with the second-order derivative of Gaussian kernel w.r.t y is denoted as: L_{yy} etc.

2.2 Edges

Edges are the most eye-catching features in images and deserve the full focus of the image processing community. An edge is defined by a sharp change in image brightness that can arise from several scenarios:

- Object boundaries e.g. a dark object on a light background or vice versa.
- Sharp changes in the surface orientation of objects e.g. the common boundary of two orthogonal planar faces of an object.
- Changes in the material properties of the viewed object e.g. changes in the reflection coefficient.
- Partial occlusions of an object by another one.
- Changes in illumination e.g. the border between a brightly lit region and a shadow region.

Mathematically an edge is a maximum of the magnitude of the first derivative in the 2-dimensional image space. This is why most edge detection methods rely on computing the gradient magnitude and performing a thresholding on the magnitude. A well accepted approach for edge extraction was proposed by Canny [24] and comprises four major steps:

- Computing of partial derivatives L_x and L_y of the image f by convolution with first-order derivatives of a Gaussian kernel g_x and g_y. The standard deviation σ of the derivative-of-Gaussian kernels determines the degree of image noise suppression: $L_x = g_x \star f, L_y = g_y \star f$.
- Computing the gradient magnitude as $m(x, y) = \sqrt{L_x(x, y)^2 + L_y(x, y)^2}$ (see Fig. 2.2(b)).

- Thresholding to find potential edge candidates i.e. locations where the gradient magnitude m is above a predefined threshold are marked as probable edge candidate (see Fig. 2.2(c)).

- Applying a non-maximum suppression scheme that filters out candidates that are not a local gradient maximum. The non-maximum suppression can also be used to calculate the location of the edgel with sub-pixel accuracy. An explanation of this extension to the classical approach can be found in the technical report of Devernay [31]. Basically the sub-pixel location is done by fitting a parabola to the magnitude sampled in gradient direction (see Fig. 2.1(d)).

- Generation of contours by tracing edges. This approach is based on the assumption that dominant edges lie on continuous curves. A hysteresis thresholding scheme involving two thresholds is applied. The higher threshold is used to select start edges for the tracing and the lower threshold is applied during the tracing and therefore accepts also fainter edges.

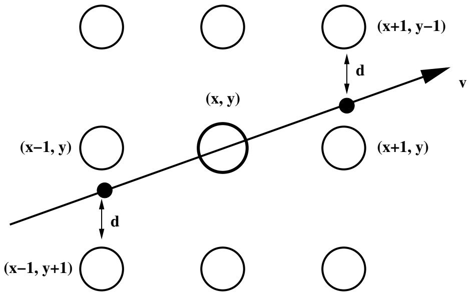
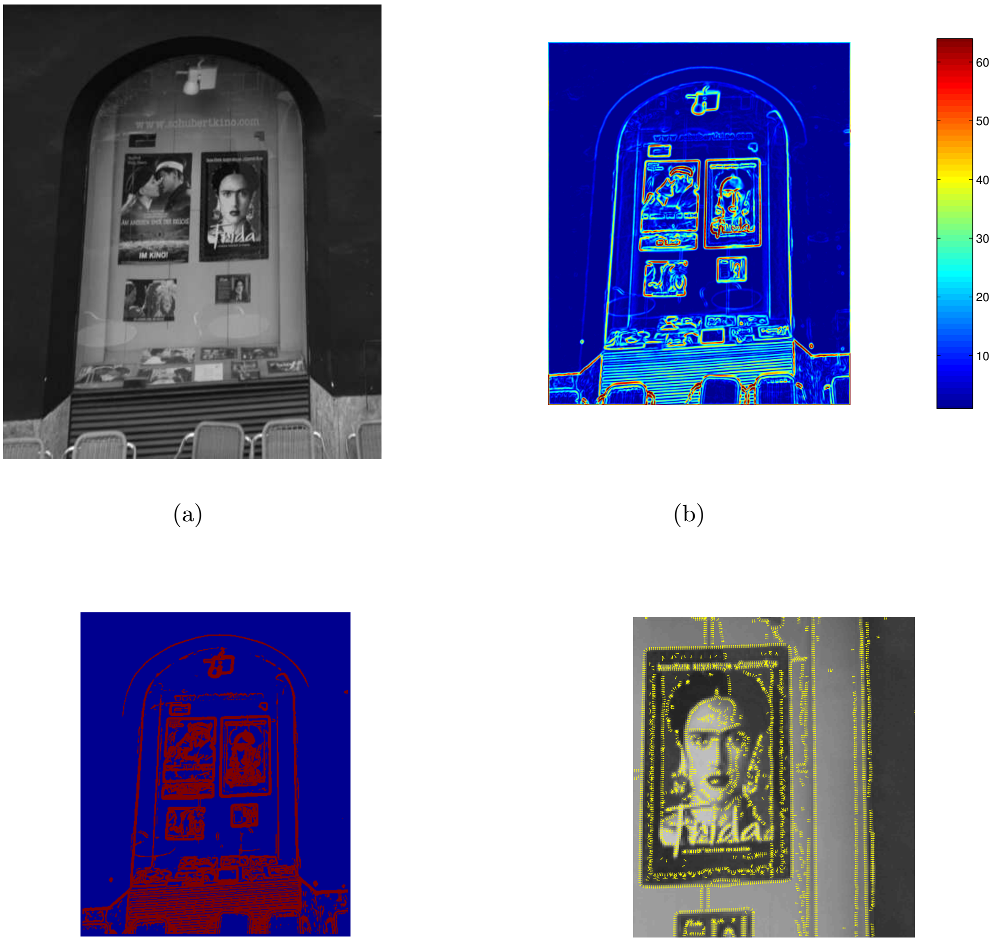


Figure 2.1: Non-maximum suppression for sub-pixel edge detection: point (x, y) is accepted as valid edgel, if the gradient magnitudes of the neighbor points sampled in the direction of the gradient direction v are smaller than the center magnitude: $m(x, y) > \max(m((x, y) + v), m((x, y) - v))$. The values at $m((x, y) + v)$ and $m((x, y) - v)$ can be approximated by: $m(m((x, y) + v) = m(x + 1, y - 1)d + m(x + 1, y + 1)(1 - d)$ and $m((x, y) - v) = m(x - 1, y - 1)(1 - d) + m(x - 1, y + 1)d$. The center value and the approximated values can be used to compute a refined maximum position by fitting a parabola.

Figure 2.2(d) shows the final edgels with their gradient direction.

The important contribution of Canny was in showing that the optimal smoothing filter can be well approximated by the first-order derivative of a Gaussian kernel. He achieved a trade-off between detection rate and localization accuracy and the proposed non-maximum suppression avoids multiple responses from a single edge.



(a)

(b)

(c)

(d)

Figure 2.2: Edge extraction example: (a) Original image; (b) Gradient magnitude; (c) All edge candidates; (d) Remaining edges after non-maximum suppression (magnified portion of the image);

An improved method for edge extraction was proposed by Rothwell et al. [106] and yields more robust results in the vicinity of junctions where the step edge model tends to fail.

2.3 Ridges

With the approach of Canny it is possible to extract step edges i.e. point locations that are on the boundary of two regions with significantly different intensities. For many applications it is also desirable to extract ridges - these are geometric entities that represent thin line-like structures in an image. Ridges can be viewed as topographical watersheds that separate image regions (sometimes denoted as *basins*). Lindeberg [121] gives a more formal definition: At an image point $p = f(x, y)$, the curvature directions p and q of the brightness function are computed. From the the second-order derivatives L_{xx} and L_{yy} and the mixed derivative L_{xy} a local coordinate system that is aligned to the principal curvature directions p and q is defined. This makes the mixed second-order derivative L_{xy} vanish. Ridges in the p, q system are then image locations where the following conditions are fulfilled:

$$L_p = 0, L_{pp} < 0, |L_{pp}| \geq |L_{qq}| \tag{2.4}$$

or

$$L_q = 0; L_q q < 0; |L_{qq}| \geq |L_{pp}| \ . \tag{2.5}$$

Depending on whether p or q corresponds to the maximum absolute value of the principal curvature. This definitions hold for bright ridges but extend naturally to dark ridges (also denoted as *valleys*). Based on the sign of the maximal absolute second-order derivatives $|L_{pp}|$ and $|L_{qq}|$ two ridge types a bright one and a dark one are defined: If $|L_{pp}| > |L_{qq}|$ and $L_{pp} < 0$ a bright ridge has been found otherwise a dark ridge (valley) has been encountered.

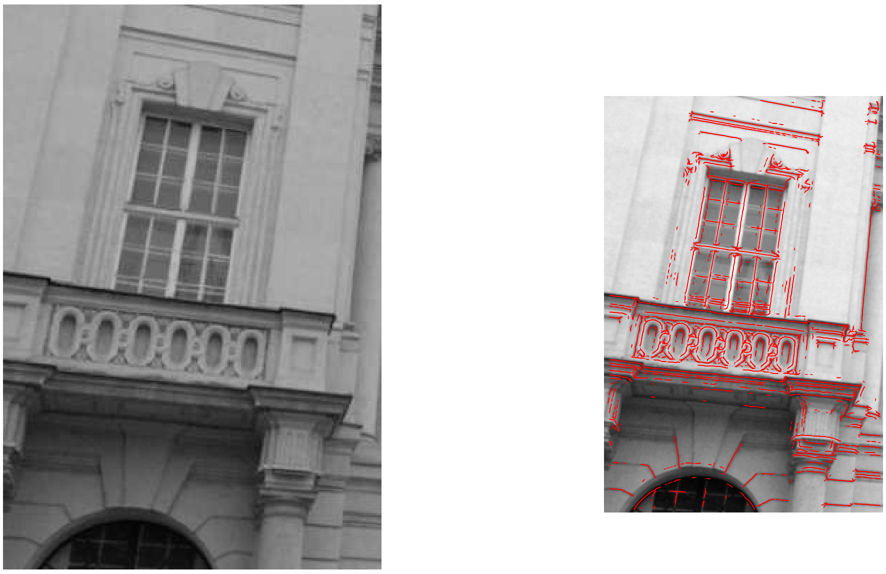
The particular ridge extraction method in this work proceeds as follows:

- Compute the the second-order derivatives L_{xx} and L_{yy} and the mixed derivative L_{xy}.

- Set up the Hessian matrix $\mathcal{H}$ for each image point: $\mathcal{H} = \left(\begin{smallmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{smallmatrix}\right)$. A ridge is detected by analyzing the eigenvalues λ_1 and λ_2 of $\mathcal{H}$: The point under consideration is a ridge or valley if:

  1. $\lambda_{max} = \max(\|\lambda_1\|, \|\lambda_2\|) > th$ (where th is a predefined threshold) and
  2. the local gradient sampled along the eigenvector corresponding to the smaller eigenvalue is below a certain threshold (this verifies the condition $L_p = 0$ or $L_q = 0$).

If the eigenvalue with the larger magnitude is smaller than zero the point is a ridge otherwise it is a valley.



(a) (b)

Figure 2.3: Ridge extraction example: (a) Original image; (b) Extracted ridgels with their associated orientation.

2.4 Corners

In many applications that are related to finding point correspondences between images one does not try to establish matches between all image pixels, but looks for points in the image that are in some way **distinct**. Such points are referred to as interest points and are located using an interest point detector. As Canny's approach is the most common method for the extraction of edges, the Harris [49] corner detector is the most popular algorithm for corner extraction from images. Corners are locations in the image where dominant and different edge directions in a local neighborhood around the point occur. This is the case when regions of different intensity meet and form distinctive shapes such as L-junctions, T-junctions or Y-junctions. One of the earliest methods for extraction corners was published by Moravec [88] and is based on analyzing the auto-correlation function for shifted templates: A small square template window (3×3 to 7×7 pixel) is placed at a position $p = (x, y)$ in the image and the intensity variation between this window and shifted instances of the window are measured. The windows are shifted into the eight principal directions: horizontally, vertically and the four diagonals. The sum of squared differences between corresponding pixels in these two windows gives the intensity variation S. For homogeneous regions the intensity variation is low, for edges it attains high values for some shifting directions and for corners S is high for all shifting directions. In the case of interest points, the auto-correlation function is high for all shifting directions. However, the operator is not invariant to rotations because the intensity variation is only calculated for a discrete set of shifting directions.

The method proposed by Beaudet [16] achieves invariance against rotation due to the fact that it derives a cornerness measure from the determinant of the Hessian matrix H which is the 2×2 matrix of the second order partial derivatives. The determinant of the Hessian $det(H) = L_{xx}L_{yy} - L_{xy}^2$ is used as a measure for the strength of interest points.

The corner extraction method from Harris and Stephens [49] is based on the second moment matrix and can still be regarded as state-of-the art. The basic idea of this detector comes from the observation that the auto-correlation function can be approximated by the second moment matrix (also called the structure tensor). The first step is the approximation of the sum of squared differences between two square windows (denoted S for the Moravec detector). This is done by constructing the tensor T that is the outer product of the 2-vector holding the partial derivatives:

$$T = \begin{pmatrix} L_x & L_y \end{pmatrix} \begin{pmatrix} L_x \\ L_y \end{pmatrix} = \begin{pmatrix} L_x^2 & L_xL_y \\ L_xL_y & L_y^2 \end{pmatrix} \tag{2.6}$$

The tensor T is of rank 1 and thus has only one non-zero eigenvalue. However, summing the tensors T_i of a local square window leads to the structure tensor T_s:

$$T_s = \sum_u \sum_v \begin{pmatrix} L_x^2 & L_xL_y \\ L_xL_y & L_y^2 \end{pmatrix} . \tag{2.7}$$

The structure tensor T_s is a positive semi-definite, symmetric matrix and thus has only positive eigenvalues. The two eigenvalues λ_1, λ_2 are used to the corner response function C:

$$C = det(T_s) - \kappa\ trace(T_s)^2 \tag{2.8}$$

The optimal value of $\kappa = 0.04$ was determined empirically. In case of corners C attains large positive values and edges are indicated by large negative values of C.

2.5 Chains

Chains are sets of connected point-like features. The method of linking points can either be straightforward as in the case of finding the border pixels of a segmented region in an image - in this case the border points can be extracted by sequence of morphological operations described in Gonzalez and Woods [45] (pp. 548). However if the points are the result of an differential geometry-based feature detection approach, this is e.g. the case for edgels or ridgels, the grouping of points to chains is a somewhat more heuristic approach. In these cases the linking is based on the analysis of properties of the feature points. The edge tracing of Canny's method is such an example, here two thresholds are used to determine initial candidates with a high gradient magnitude and the contour following is terminated if a new candidate points falls below a lower threshold. Properties under consideration are in this case the proximity (only immediate neighbor pixels are examined) and the gradient magnitude due to the hysteresis thresholding approach. A brief sketch of a general linking method can also be found in [45] (pp. 585).

The introduction of further constraints can make the contour linking more stable. Typically the linking strategy should also incorporate the angular difference between candidate points and also be able to bridge gaps. In the following we propose a general method for linking oriented point features (typically edges or ridges). The proposed linking approach no longer works on the discrete image grid but directly on the subpixel coordinates of the extracted features. Thus the linking of points to chains involves the search for the nearest neighbors around a query point (the last point added to the contour chain). In order to avoid the exhaustive search through the whole set of extracted points, a KD-tree is used for efficient range queries. This data structure allows to detect points in a local search range around a query location in $O(\log(n))$.

These features have a principal direction associated to them, which can be used to make the chaining process more robust. A simple growing algorithm for the generation of contour chains is the following (see Figure 2.4 for an illustration):

1. Choose two values th_{high} and th_{low} for the hysteresis-based thresholding.
2. Sort extracted features according to their magnitude m (gradient magnitude for edgels, absolute curvature for ridgels). This sorting avoids that the chain growing starts at points with a magnitude below th_{high}.
3. Build a 2D KD-tree for the feature positions - this allows for fast nearest neighbor queries.

4. Select a seed point p_s with a magnitude $m \geq t_{high}$.

5. Start the growing from p_s and accept only features that lie within a predefined search radius r, have a magnitude $m \geq t_{low}$ and have the same general orientation as the current terminal point of existing chain (v) i.e. the inner product of the directions must be greater than zero. At the start of the growing this is the direction of the seed point, if more points are inserted the direction can be estimated from the most recently inserted points of the contour chain. In case of ridge linking the type, either ridge or valley, is also taken into account. Every inserted point is marked as invalid.

6. If no more points can be added, the growing is repeated at the starting point p_s, in the opposite direction.

7. If no more points can be found for the current chain, select a new seed point and continue until no more valid seed points are available.

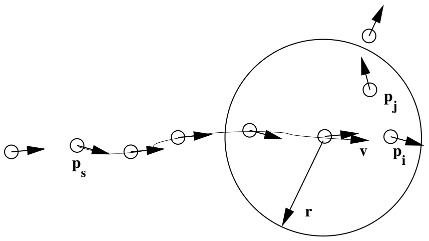


Figure 2.4: Contour growing scheme. Starting from a seed point p_s, points with a sufficiently small angular difference and a distance below r are added to the contour. While the enclosed angle between point p_j and the current chain direction v is too high, point p_i fulfills the angle and distance criterion and is therefore added to the chain.

In figure 2.5 extracted edgel and ridgel chains are shown.

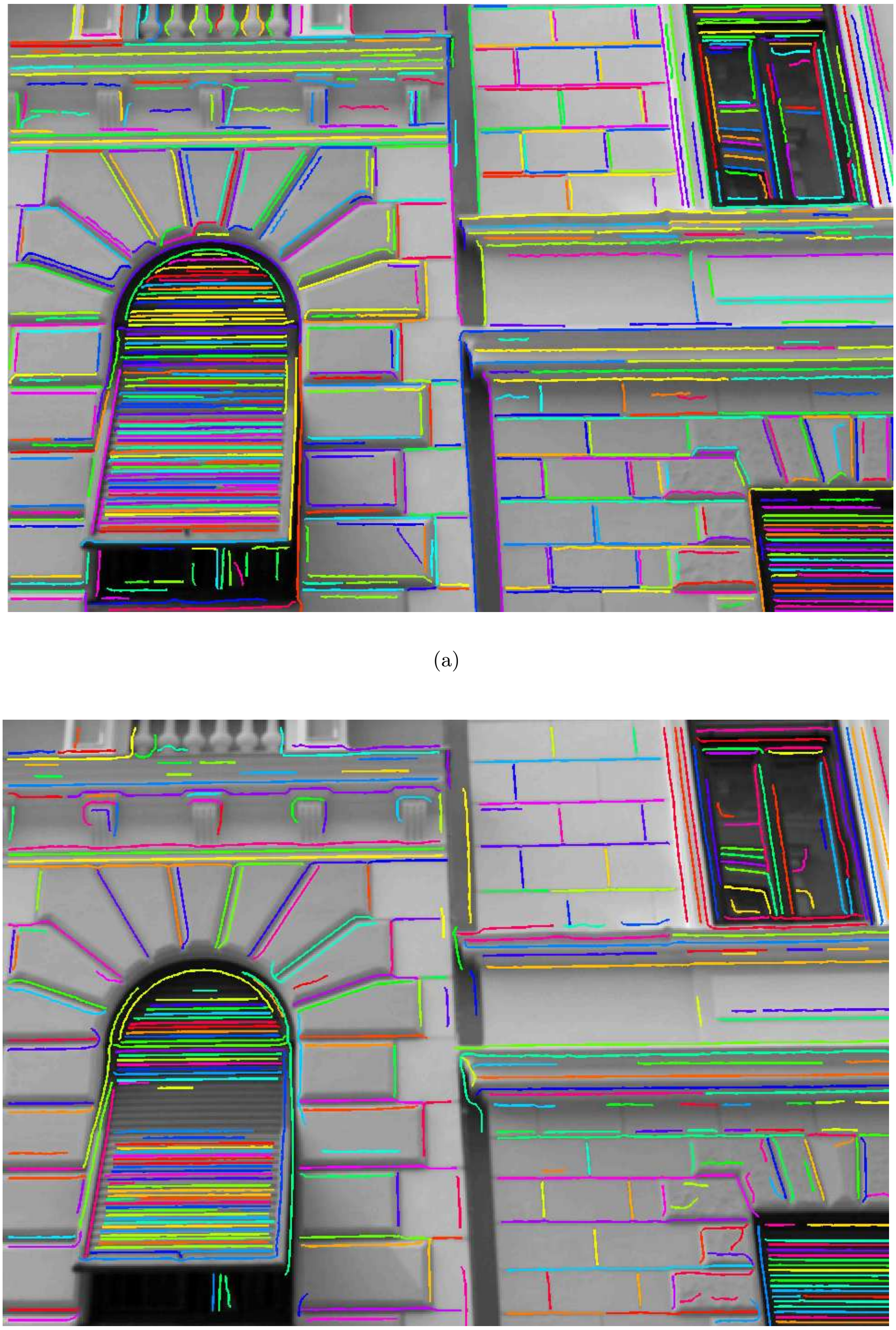


Figure 2.5: Illustration of contour chain extraction from edgels (a) and ridgels (b). The image has a resolution of 491 × 338 pixel. Note that the fine structure of the shutters is captured well in both plots.

2.6 Lines

Straight lines are among the most prominent features in man-made scenes. Their extraction has been the subject of extensive studies since the early years of image processing.

2.6.1 Hough transform

A popular method for line extraction was published by Hough [54]. The so called Hough transform detects straight lines in 2D point sets by transforming the points to the line parameter space, where each point represents a pencil of lines. The transform to the bounded line parameter space allows to define a discrete, and thus memory efficient, accumulator space. In this accumulator space groups of collinear points form local maxima. The local maxima in the accumulator represent the parameters of potential straight 2D lines. In its original definition the HT is a global method with the following properties:

- The Hough transform is an efficient method for detecting dominant lines.
- Few parameters are necessary to control the algorithm.
- The Hough transform lacks of locality: short segments are likely to be missed due to the fact that their maxima disappear in the noise produced by spurious responses from groups of non-connected, collinear points.
- The original Hough transform does not make use of direction information (e.g. gradient vectors) associated to the points. The integration of direction information can increase the robustness and speed of the method.

Several improvements of the HT have been published and the method has been extended to detect other parametric primitives, but due to the fact that the detection of parametric shapes with n parameters requires an n-dimensional parameter space Hough transforms are limited to detect simple shapes such as circles (3 degrees of freedom) and ellipses (5 degrees of freedom). Illingworth [57] gives an extensive overview on the various HT techniques.

2.6.2 Local methods

Local methods for the extraction of straight lines can be found in [23, 91, 132]. The Burns line finder [23] is designed to detect straight lines in complex images of outdoor scenes and can be considered as the prototype of local line extraction methods. The method is split up into four basic steps:

1. Firstly, edgels are grouped according to their gradient orientation.
2. The surface defined by the image's intensity values is approximated by a weighted least squares fit.
3. The line parameters are extracted from the result of the fitting step.
4. A filtering stage selects the desired candidates based on various criteria (minimal length, minimal contrast, line groups with a particular orientation etc.)

The general observation for local line extraction methods is that more parameters are necessary to control the algorithms. This makes the approaches somewhat more heuristic than the global Hough transform.

Lately Schmid and Zisserman [111] proposed an approach to solve the line extraction problem by connecting the Canny edgels to contour chains (similar to the algorithm described in 2.5) and then fitting lines to this chains.

In the following two approaches for detecting straight 2D line segments are presented. The detailed outline of the approaches is intended to familiarize the reader with the general concept of line detection for 2D point sets. The first approach directly operates on directed point-like primitives such as edgels or ridgels. The second method uses contour chains as input. Both approaches are capable of extracting robust line segments in an efficient manner.

Line detection for directed, point-like 2D primitives The first algorithm takes a set of edgels as input. The edgels are sorted according to their magnitude in descending order. The sorting yields good starting points for the line segment detection, since edgels that are close to corners have a lower magnitude. A KD-tree is build for the Canny edgel set - this allows efficient range queries on the point set. In order to detect line hypothesis, a bottom-up approach is used. Each edgel with its position $p_s = (x, y)$ and its tangent direction $t = (-dy, dx)$ (perpendicular to the gradient direction $g = (dx, dy)$) defines a line hypothesis. The validity of the hypothesis is verified considering all neighboring points within distance r around the point (see Figure 2.6) and is proportional to the number of supporting points. A supporting point must fulfill several criteria: It must lie within the maximal perpendicular distance d of the line hypothesis (solid lines), the enclosed angle of its tangent direction and the lines direction must not exceed a certain threshold φ_{max}. The *score* of a line hypothesis is then computed as the number of supporting points, divided by the search radius r thus representing the density of points in the search region. Typical parameters are $d = 0.3...1.5pixel$, $r = 3...10pixel$, $\varphi_{max} = 20...90°$ and $score_{min} = 0.5...1.0$. Every line hypothesis with a $score > score_{min}$ is then processed further. All accepted hypotheses are then handed over to a growing process, similar to the one described in 2.5, where new edgels are added based on distance and enclosed angle thresholds.

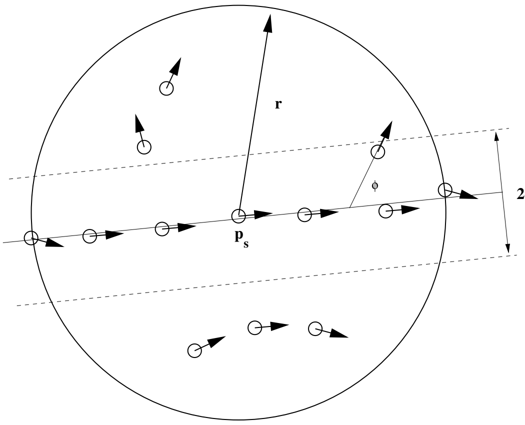


Figure 2.6: Hypothesis verification for a line segment. The tangent direction of a seed point p_s defines a 2D line hypothesis. In order to support the hypothesis, a directed point must lie within the search radius r, have a perpendicular distance smaller than d and an enclosed angle smaller than ϕ.

Line detection for contour chains The second approach for line segment extraction is inspired by [111] and based on the analysis of contour chains. Since the complexity of contour chains is higher than that of an edgel set, the algorithm works with fewer parameters and due to the fact that a set of connected points is analyzed, the verification of line hypothesis is significantly faster. The algorithm works as follows: For every chain the best start segment (with a maximal number of collinear edgels) is searched by calculating a least squares line fit to a local subset of contour points and summing up the perpendicular distances as a quality measure. If a valid start segment is encountered the algorithm tries to increase the length of the line segment by subsequently adding new points. This growing is constrained by a maximal gap that can be bridged to a new point and the enclosed angle of the point's tangent direction and the line's direction. Figure 2.7 shows a typical scenario, where the initial segment is the dashed rectangle and the solid rectangle is the final line segment after the growing step (the width of the box symbolizes the maximal perpendicular distance).

Figure 2.8 shows the extracted 2D lines segments for a subpart of a facade image.

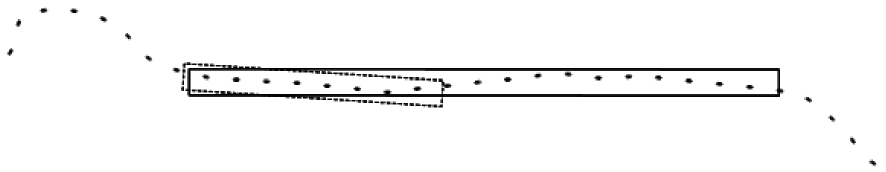
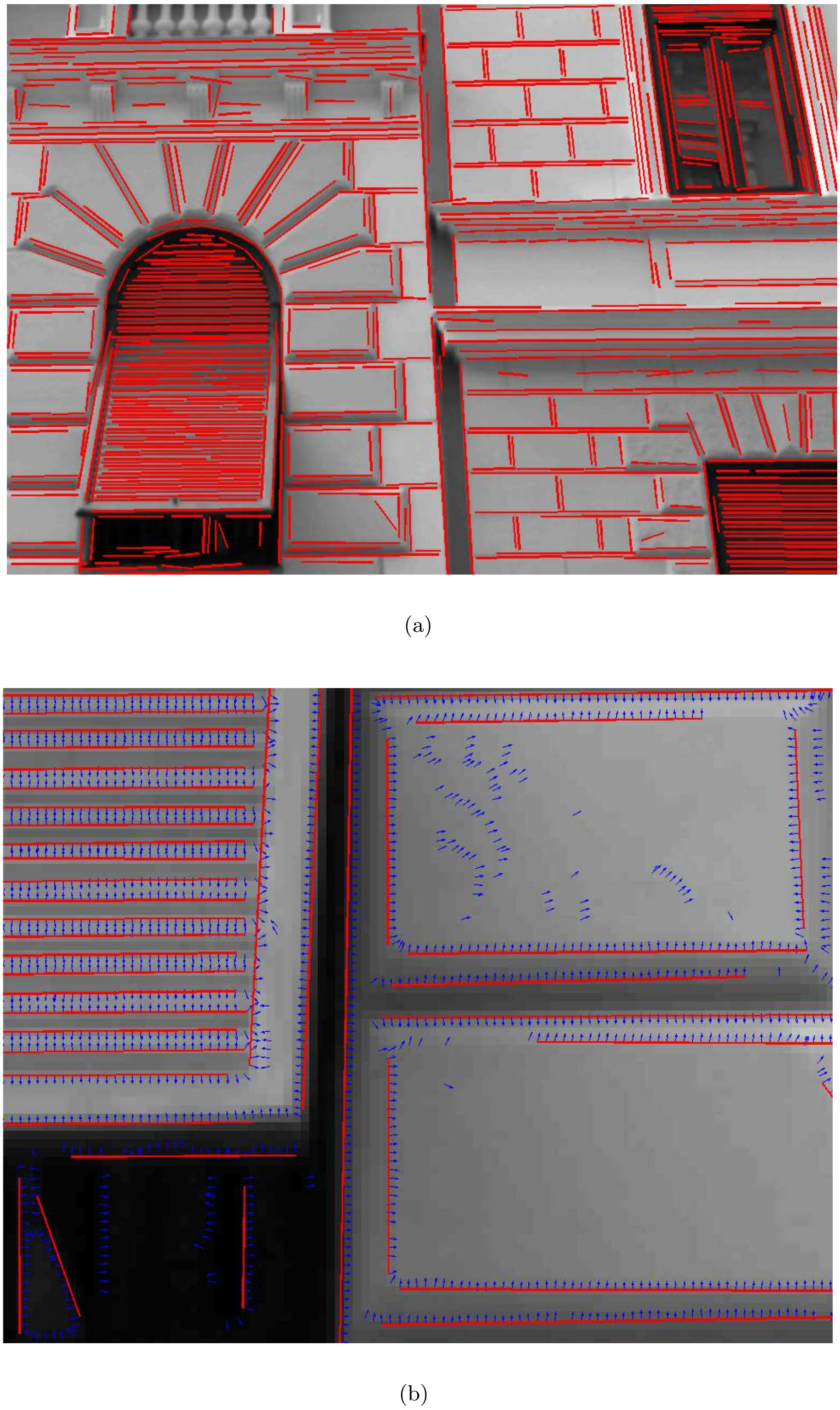


Figure 2.7: Growing of a line segment for a contour chain. The points of the initial line segment define the orientation and length of the search region (shown as dashed rectangle). After adding new points the new search region (shown as solid rectangle) is longer and has a slightly different orientation. The width of the box symbolizes the maximal perpendicular distance.



(a)

(b)

Figure 2.8: Illustration of straight 2D line segment extraction from edgels. Plot (a) shows an overall view with the line segments drawn in red and (b) a zoomed out detail view with the lines in red and the edgels (symbolized with their gradient direction) in blue. The image has a resolution of 491 × 338 pixels. Note that even for the fine structure of the shutters many line hypothesis are detected.

2.7 Vanishing points

Buildings, as well as other man-made structures, exhibit many linear structures. Furthermore the linear structures are often in a parallel arrangement. An important property of parallel linear structures is that, in perspective mapping, those linear structures converge in a single point, called the vanishing point. Vanishing points have been used since the Renaissance when artists concluded that lines that are parallel in 3D, when properly transferred to the image, would appear to meet at a single point on the horizon. A typical scenario is shown in Figure 2.9: The parallel lines of the depicted facades are converging to three different vanishing points (illustrated in red, green and blue).

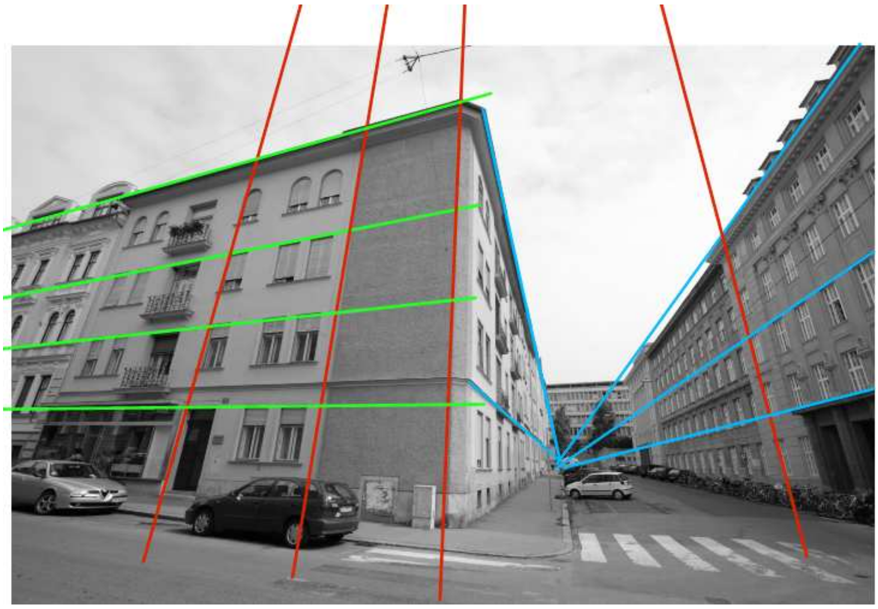


Figure 2.9: Typical vanishing point scenario in an urban environment. Three dominant vanishing points are present in this image: The vertical vanishing point (read lines) and two horizontal vanishing points (green and blue). One horizontal vanishing point is located in the image.

2.7.1 Vanishing point extraction

In literature many methods for automatic vanishing point extraction have been published. In the following the most prominent methods will be introduced and their advantages and disadvantages will be discussed.

An efficient approach for vanishing point detection was proposed by Kosecka and Zhang [70]. Their detection process relies on a two stage strategy, where in the first stage, line segments are extracted and in the second stage the location of the vanishing points is determined by an expectation maximization (EM) scheme.

In the method proposed by Tuytelaars et al. [128] vanishing points are detected by repeatedly applying the Hough transform: the first run of the Hough transform uses the location of extracted edges as input and therefore detects straight line hypothesis.

In the second run the transform is applied on the most prominent peaks from the first run (basically a parametric representation of detected lines) and detects locations where several straight lines in the original image intersect. In order to map line the intersections to a bounded accumulator space, different parametric representations are chosen for the areas outside the original image bounds. This method has the advantage that it works on point primitives and needs no additional information about the camera's intrinsic geometry (focal length, principal point).

Another accumulator-based method for finding vanishing points can be found in [27]. This approach assumes that the intrinsic camera parameters are known. With the known camera geometry it is possible to compute the planes that are generated by the two rays that correspond to the line's end-points. If these planes are now intersected with a Gaussian sphere (sphere with unit radius) that is placed on the camera's origin, the resulting intersections are great circles on the sphere. This is explained by the fact that all rays (and therefore all planes) also pass through the camera's origin. The method of intersecting planes with a sphere allows to transform the line intersections from the unbounded image plane onto the bounded surface of a sphere. In this case the accumulator cells are the triangles of the tessellated sphere.

Schaffalitzky and Zisserman [108] use a RANSAC technique to detect the vanishing points but also statistical methods as in [131] are used for detection.

In the following two methods for vanishing point detection will be discussed in more detail. These two methods use very different approaches to achieve the same goal and therefore reflect the great variety of available methods.

Rother's method Rother [103] published a method, that directly works on the extracted line segments and is based on a voting scheme for the detected intersections of line segments. In Rother's approach, vanishing points are detected by applying a Hough-like algorithm where the image plane is directly used as accumulator space. The input data for the algorithm are line segments that are extracted in a pre-processing step. For all extracted line segments in the image, the mutual intersections are computed. These intersections are then used as accumulator cells for the detection process. Each intersection is treated as a *potential vanishing point* and votes for each cell are determined by inspecting all other line segments. Figure 2.10 shows a line segment s of length l with the midpoint m. The weight w of the line segment s for the intersection point i formed by the line segments l' and l'' is calculated as follows:

$$w(s) = k_1 \left(1 - \frac{\alpha}{\alpha_{max}}\right) + k_2 \left(\frac{l}{l_{max}}\right) \tag{2.9}$$

Where α_{max} is the maximal enclosed angle between the line segment and the vector pointing to the intersection - if a line segment exceeds this maximal angle it is not taken into account for the voting of the respective intersection; l_{max} is the length of the longest line segment and k_1 and k_2 are set to 0.3 and 0.7 respectively (according to Rother). The final formula for the voting process of all line segments in the set L for an intersection point is:

$$W(i) = \sum_{\forall l \in L} w(l) | \alpha \leq \alpha_{max} \tag{2.10}$$

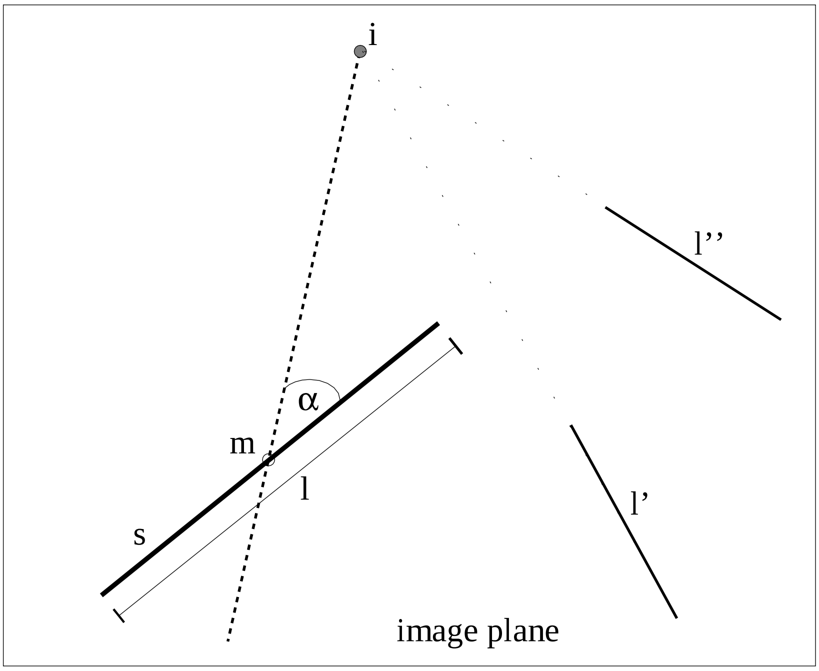


Figure 2.10: Orientation of a line segment s versus the direction towards a potential vanishing point. The contribution weight of the segment is computed from its length l, the enclosed angle α between the line vector and the vector from the line segments mid point to the vanishing point i.

This voting process is applied to all intersections and the intersection with the highest weight is then accepted as potential vanishing point.

Thales point method Brauer-Burchardt and Voss [21] propose a method for determining vanishing points in images based on the Thales circle method (TCM). Figure 2.11 illustrates the principle: An arbitrary reference point p_r is chosen (it may lie anywhere within the 2D plane but typically within the image borders) and based on the assumption that every line belongs to a vanishing point the Thales points (TP's) are computed. This is performed for all lines within the image. The result of this transform is that, instead of dealing with intersection points that lie in the unbounded 2D plane, a set of points within the image bounds is created. All TP's that belong to a specific vanishing point lie on a circle - the Thales circle. In their experiments Brauer-Burchardt and Voss used a set of synthetic line segments, that point towards a single vanishing point.

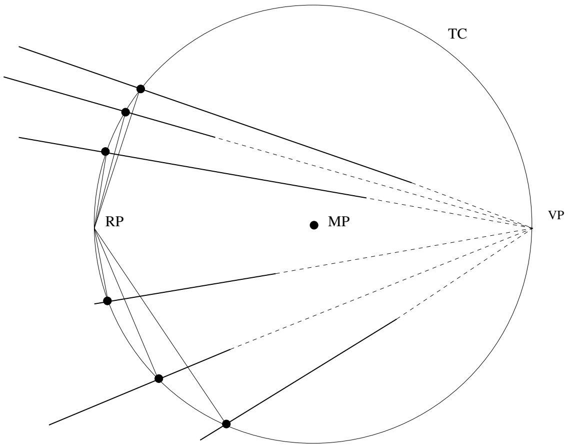


Figure 2.11: Vanishing point detection using Thales' theorem. If a number of line segments point towards a common (vanishing) point VP, the orthogonal projections of an arbitrary reference point RP onto the line segments (illustrated as black dots) are points lying on a circle - the Thales circle TC.

However for practical applications the line segments that are detected in an image will belong to more than one vanishing point. A typical facade image exhibits line segments that belong to two or three vanishing points and there may also be a number of line segments that can not be assigned to any vanishing point. Furthermore lines belonging to a vanishing point will be corrupted by image noise.

In the following an extension of the TCM that allows the robust detection of multiple vanishing points is presented. Due to the presence of multiple vanishing points and the corrupting influence of image noise the Thales points for the individual vanishing points will form rather noisy circles. The key to robust detection of vanishing points is an outlier tolerant circle detection method based on the RANSAC principle: Three points are randomly chosen and a circle hypothesis is computed from these three points. All TP's are now tested for their distance to the circle hypothesis. If a TP distance to the circle is below a predefined threshold distance it is considered to be an inlier for the current vanishing point hypothesis. In contrast to general circle detection, where only 2D points are available, the TP's also provide directional information (from their generating line segment). This directional information can be used as an additional criterion in inlier detection. Figure 2.12 shows how the directional information can be used for putting a further constraint to inlier selection: For a circle formed from three randomly chosen TP's the location of the vanishing point VP can be inferred from the midpoint MP. The inliers are selected from those TP's with a sufficiently small distance from the Thales circle TC and with a sufficiently small enclosed angle between the vector that is formed by the line segment's mid point and VP and the line segments vector l_v. In the illustration line segment l_3 would be accepted as inlier because its TP lies within the distance threshold from the circle and its enclosed angle α_0 is also below a given threshold, line segment l_4 would be rejected for not fulfilling the enclosed angle criterion.

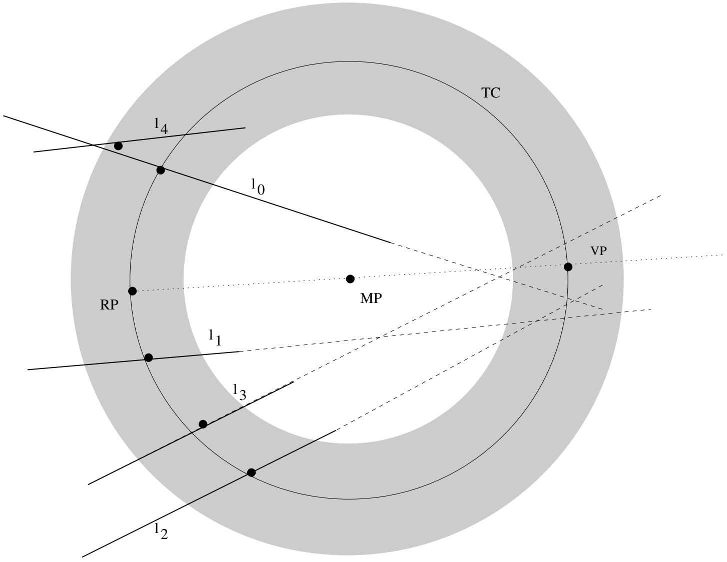


Figure 2.12: Inlier selection for the Thales circle detection. Three randomly chosen Thales points, belonging to the line segments l_0, l_1 and l_2 form the Thales circle TC. Inlier points must lie within the gray shaded area that defines the maximal perpendicular distance of a Thales point to the circle and must also point to the vanishing point VP.

In the experimental section it will be shown that the proposed extension is capable of detecting multiple vanishing points in a robust manner. Figure 2.13 illustrates the Thales circle detection for a typical urban vanishing point scenario. Extracted 2D line segments (in gray), the corresponding Thales points (black dots) and the three detected Thales circles are shown. The RANSAC-based circle detection is robust against outliers and the circles radii, especially for the large circles, are detected robustly, despite the low coverage (the largest Thales circle has a point coverage of approximately 15% of its circumference). The original image is shown in figure 2.14.

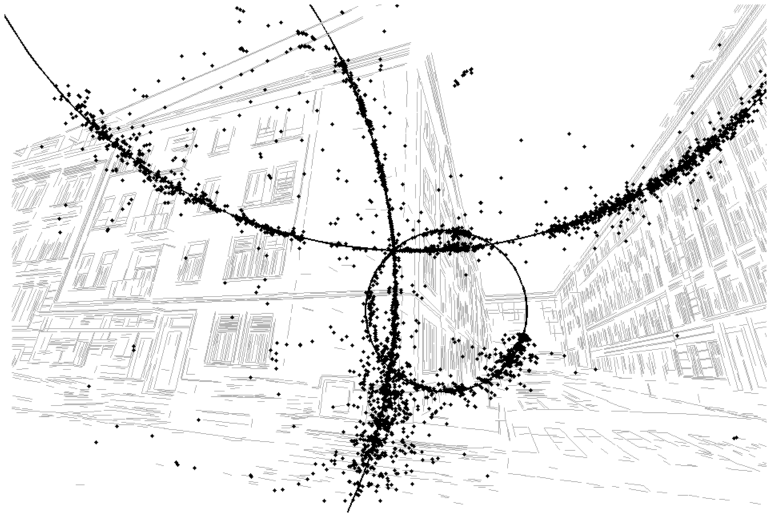


Figure 2.13: Vanishing point detection using the Thales circle method. The plot illustrates extracted 2D line segments (in gray), Thales points (as black dots) and the three Thales circles that correspond to three dominant vanishing points in the image. The image has a resolution of 2032 × 1352 pixel and the reference point is in the center of the image.

Figure 2.14 shows an example of vanishing point extraction with the Thales circle method.

A comparison showed the proposed method is superior to Rother's method in execution speed, and reaches equal levels of robustness and accuracy. The main reason for lower running times lies in the amount of data that have to be processed: The number of Thales points on which the RANSAC-based circle detection is performed, is the same as the number of input lines, whereas Rother's method creates all pairwise line intersections which results in approximately $\frac{n^2}{2}$ intersections for n input lines. In practical applications the vanishing point detection methods report vanishing point hypotheses and the corresponding line segments and the optimal vanishing point location is determined by a least squares solution.

In this refinement step the detected location of the vanishing point is improved using the maximum likelihood estimate (MLE) procedure proposed by Hartley and Zisserman in [108]. In this approach the distance between the end points of line segments that vote for a vanishing point and the line that goes through the mid point of the voting segment to the vanishing point is minimized using a nonlinear least squares optimization method.

2.7.2 Applications using vanishing points

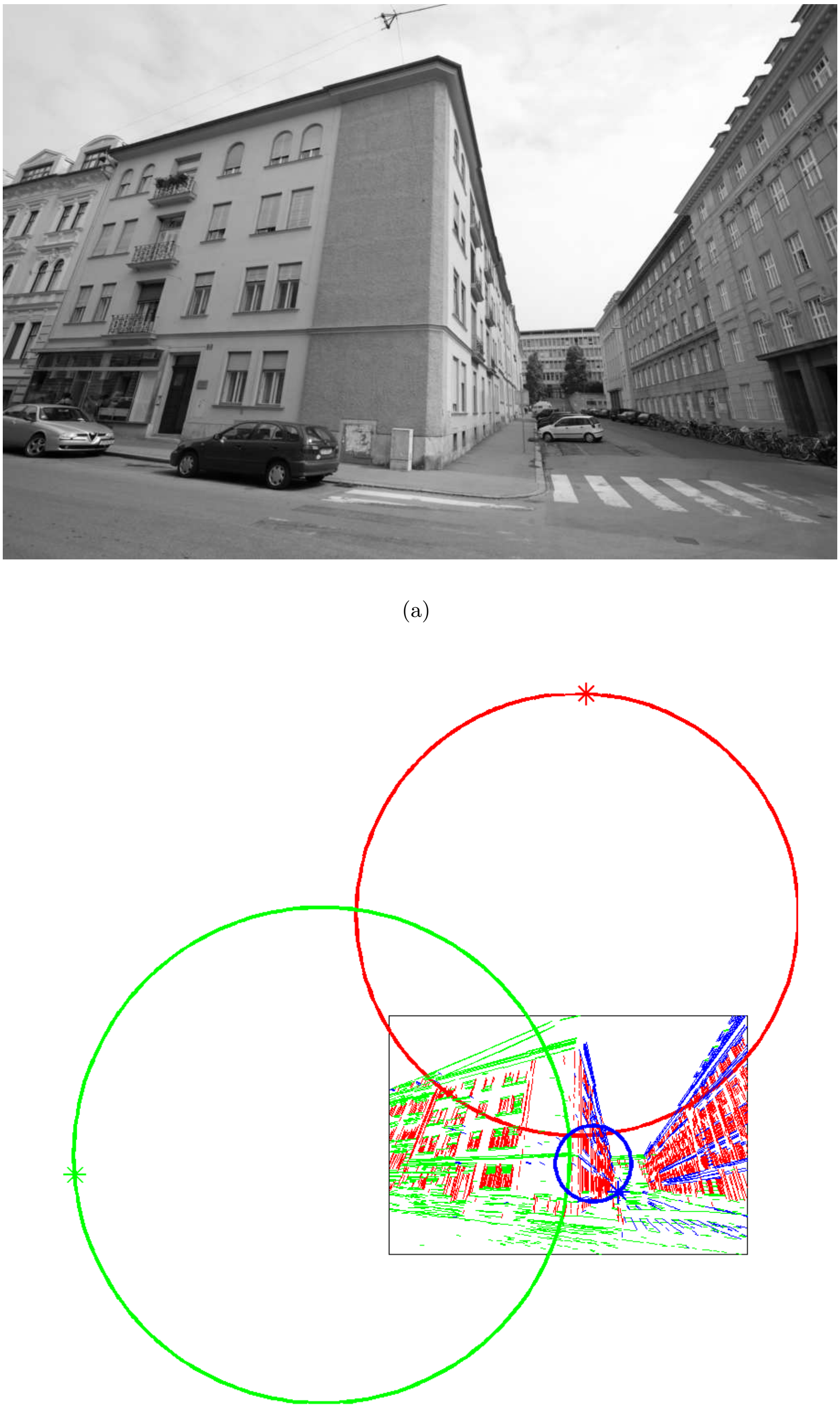
Vanishing points provide information about the scene structure and can be used to detect plane hypotheses, perform a rectification and in the case of calibrated cameras the presence of orthogonal vanishing points can be used to compute the rotation between the camera and the observed planar structure: If two orthogonal vanishing points have been detected within an image, the relative rotation of the (calibrated) camera with respect to the vanishing directions can be computed. Using this relative rotation it is possible to perform a so called rectification of the image. Since the camera undergoes a pure rotation for the rectification step, this transform can be expressed as a 3 × 3 homography matrix. Figures 2.15(a) and (b) show an exemplary rectification for an image of a facade plane.

A logical extension is the use of orthogonal vanishing point pairs for estimating the relative rotation between image pairs.

If only one vanishing point can be detected in an image this information can be used to define a reference direction within the image. This reference direction can then used to compute rotational invariant descriptors, thus making the point correspondence detection an easier task.

If two orthogonal vanishing points can be detected, their geometric configuration can be used to determine the principal point of the camera (see [50] pp. 226). Given the knowledge of three orthogonal vanishing points a camera calibration can be performed. (see also [50] pp. 226).

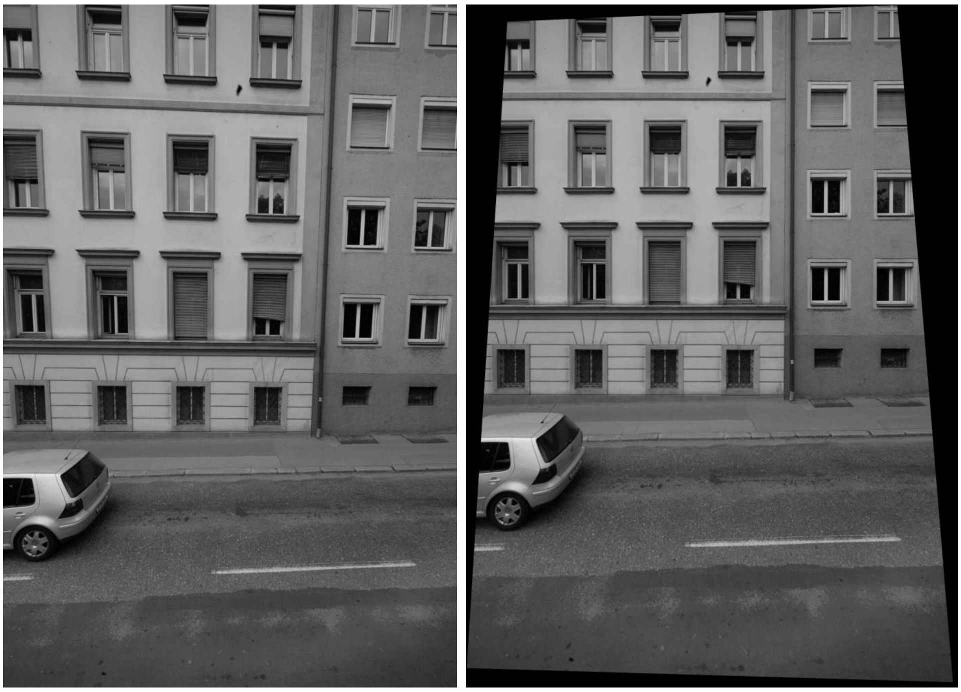
In the experimental section the method of Rother and the proposed extension to the method of Brauer-Burchardt and Voss will be analyzed for their usability for vanishing point detection in man-made scenes.



(a)

(b)

Figure 2.14: Illustration of vanishing point extraction with the Thales circle method. Plot (a) shows the image and (b) shows the line segments that correspond to the three extracted vanishing points (in red, green and blue) and the detected Thales circles. The location of the vanishing point is marked by a star symbol on the circles.



(a)

(b)

Figure 2.15: Rectification of an image, based on orthogonal vanishing points. (a) shows the original image captured by a mobile mapping setup, (b) shows the result of image rectification. The rectification removes perspective distortion for all features on the facade plane. Since the transform is a pure rotation the aspect ratio is left unchanged.

2.8 Circles and ellipses

From the class of conics $ax^2 + bxy + cy^2 + dx + ey + f = 0$ that comprises parabolas, hyperbolas and ellipses, the ellipse is the shape for which many detection and fitting algorithms have been proposed. Ellipses belong to the family of conics and are described with five parameters either explicitly via center, large axis, small axis and rotation angle, or the implicit equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{2.11}$$

One of the properties that makes the ellipse particularly useful is the fact, that the perspective projection of a circle always is an ellipse. Furthermore is the perspective projection of an ellipse again an ellipse (see [50] page 36).

2.8.1 Ellipse properties

Elementary properties of ellipses defined by equation 2.11 are:

- **Area:** πab
- **Eccentricity:** $e = \sqrt{1 - \frac{b^2}{a^2}}$. e varies from 0 (circle) to 1 and reflects the elongation of the ellipse.
- **Circumference:** The computation of the exact circumference leads to an elliptic integral, but the Indian mathematician Ramanujan found a good approximation: $c \approx \pi[3(a+b) - \sqrt{(3a+b)(a+3b)}]$

2.8.2 Ellipse detection/fitting

The Hough transform is among the most widely used methods for detecting ellipses, but since the ellipse is described by five parameters (center, large axis, small axis and rotation angle) the parameter space has considerable memory demands and makes the HT computationally expensive.

An alternative approach, described by Cheng [25], uses a modified version of the RANSAC [33] technique that does not depend on global thresholds (K-RANSAC). The proposed method works on labeled edge images and thus does not make use of directional information associated to the edges.

Fitzgibbon and Fisher [34] published an extensive comparison of the various ellipse fitting methods in terms of the used error measures and analyzed the behavior of the fitting under different levels of noise and occlusion.

In order to increase the robustness of ellipse detection and to lower the computational complexity, ellipse detection is performed on higher level primitives such as contour chains. As Fitzgibbon et al. [34] showed in their experiments, the contour must cover at least fifty to sixty percent of the ellipse's circumference to guarantee a robust fitting result, given moderate noise levels. Another problem with ellipse fitting is, that unconstrained fitting often results in one of the two other conics, namely a parabola or a hyperbola. Pilu, Fitzgibbon and Fisher [35, 93] solved this problem by introducing a direct ellipse specific fitting method that always returns coefficients that describe an ellipse.

Since it can not be ensured that a chain that possibly forms an ellipse, contains points lying on the ellipse only, it is necessary to deal with possible outliers. The RANSAC technique [33] is known to perform robustly in presence of outliers. A strategy for the robust detection of ellipse hypotheses draws samples of five random points, computes the ellipse specific conic parameters and tests the hypothesis using all points of the contour. If an ellipse is detected for a sample of points, hypothesis is improved by computing a least squares fit for all inlier points. Figure 2.16 shows the result of this the ellipse detection/fitting process on contours extracted in an image of a facade.

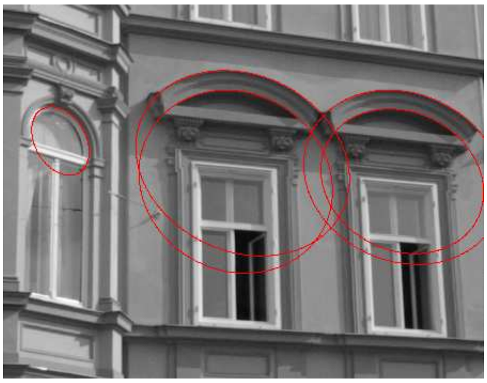


Figure 2.16: Ellipse fitting result for a part of a building facade. Ellipses are detected for the arches on top of the windows.

2.9 Affine transformed primitives

Many structures in urban environments are composed from rectangular shapes. So it seems quite natural to develop methods that can detect these shapes. However, due to the perspective projection rectangular shapes are mapped to general quadrilaterals. Thus, a general approach for detection will have to estimate not only the parameters of the rectangular shape, but also the perspective transform that maps the physical points into the image space. Furthermore, a robust approach will have to cope with noisy data and outliers as well. These observations make it clear that the robust detection of rectangular shapes is a challenging task. Several approaches for detecting rectangular shapes in urban environments have been proposed. Kosecka and Zhang [71] propose the use of line segments and vanishing points to reduce the complexity of the problem. The presence of two orthogonal vanishing points is used to generate rectangle hypotheses by intersecting line segments belonging to different vanishing points. The hypotheses are then verified in a two step approach: first by searching for supporting corner points for the rectangle corners and second by computing a normalized fronto-parallel patch for the hypothesis region and analyzing the image orientation (which should be consistent with the vanishing direction). The main disadvantage of this approach is the exhaustive search for rectangle hypothesis in the extracted line segments. A more efficient method was recently proposed by Micusík et al. [81] which also works on line segments and dominant orthogonal vanishing points. The detection of rectangles is achieved by searching the Maximum Aposteriori Probability (MAP) solution of the Markov-random-field defined on line segments that are consistent with the vanishing points. Both methods have in common that they avoid the estimation of the perspective distortion by using orthogonal vanishing points.

In contrast to these approaches stand the local methods that infer also the perspective mapping. These methods generally work on 2D point sets that are extracted from the images. A primitive that is regarded as well suited for describing rectangular shapes is the superellipse.

2.9.1 Affine super ellipses

By adding the parameter ϵ the class of ellipses can be generalized to describe a much wider range of shapes. Equation 2.12 describes a general axis-aligned superellipse with sides lengths a and b.

$$\left(\frac{x}{a}\right)^{\epsilon} + \left(\frac{y}{b}\right)^{\epsilon} = 1 \tag{2.12}$$

The parametric representation is:

$$x = \pm a \cos^{\frac{2}{\epsilon}} t \tag{2.13}$$

$$y = \pm a \sin^{\frac{2}{\epsilon}} t \tag{2.14}$$

$$\tag{2.15}$$

for $t = 0 \ldots \pi$.

Figure 2.17 shows super ellipses (ASE's) generated with different values of ϵ and $a = b$. With values for ϵ varying from $0.2 \ldots 7$ the shape changes from a pinched diamond, sometimes also named a hyper ellipse, at $\epsilon = 0.2$ to a circle ($\epsilon = 1$) and gradually to an axis aligned square ($\epsilon \rightarrow \infty$).

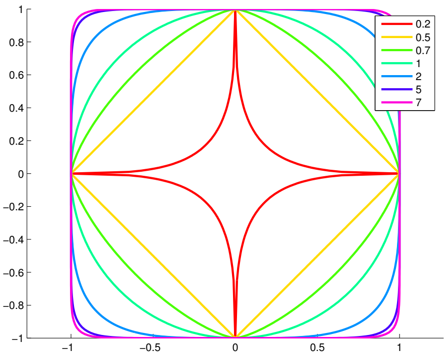


Figure 2.17: Super ellipses for varying parameter ϵ
Super ellipses for $\epsilon = [0.2, 0.5, 0.7, 1.0, 2.0, 5.0, 7.0]$ and $a = b$.

This great variety of shapes makes the super ellipse interesting for shape description.

Especially the diamond and the square are shapes that often occur on facades and other man-made structures. However since the projective mapped versions of squares or rectangles seldom result in the exact shape of the super ellipse, a further generalization is necessary. Osian et al. [92] proposed the use of an affine transform to fit super ellipses to partial contours. They also introduced a simplified error measure that allows for an efficient evaluation of the fitting error.

Since the equation for the super ellipse is no longer linear, the process of detection/fitting involves iterative, nonlinear optimization processes. Due to the high number of degrees of freedom, an efficient detection with RANSAC is not possible. Rosin [102] and Zhang and Rosin [139] describe methods for fitting super ellipses to complete and partial contours, and perform evaluations for different approximations of the error-of-fit. Their conclusion is that for partial contours and in the presence of outliers, a full 6 degree of freedom fit performs best. The error-of-fit that Osian chose, approximates the distance of a data point to the super ellipse by taking the distance of the ray that goes from the center of the ellipse through the ellipse boundary to the data point.

2.9.2 Affine transformed squares

Images of urban surroundings are typically composed of many linear structures and on a higher level the linear structures group into rectangles, triangles or polygons. Rectangles are by far the most dominant structure in facade images. Nearly every window, door, advertisement etc. is of rectangular shape. However due to the perspective imaging process the orthogonality is not preserved and the rectangles are mapped as general convex polygons. If the rectangular structures are small with regard to the image size the perspective distortion can be approximated by an affine transform. In the following a method for fitting affinely transformed squares in noisy 2D point sets is presented. The point sets are generated by segmenting the image into regions and extracting the region's outer contour. In our study we use MSER regions as proposed by Matas et al. [61], but any other region segmentation method can be used. Since the segmentation process extracts many contours from which only a subset represents perspective distorted rectangles and furthermore the contours of this subset are contaminated by noise and outliers (points that do not belong to the squares/rectangles outline), a robust fitting algorithm has to be applied.

The proposed method uses a nonlinear optimization framework that estimates the affine transform that maps the contour points to an axis aligned canonical square (unit side length, centered at the origin). Using this primitive instead of to the previously described affine super ellipses makes the method more stable against noise. Especially the parameter ϵ that controls the shape of the superellipse is sensitive to noise. Given the fact that we are looking for squares or rectangles this restriction is no limitation.

The optimization criterion is the minimization of the sum of squared distances of the 2D points to the sides of an affine distorted square. Thus the goal of the fitting process is to find the six parameters of the affine transform A (x/y-translation, x/y-scale, rotation and shear) that define the optimal transformed square for the given set of 2D points.

However, for the efficient computation of this point-to-affine square distances an inverse mapping is used. This means that instead of transforming the unit square to match the point set, points are mapped to the unit square. As a consequence the parameters of the inverse affine transform A^{-1} are optimized. The advantage of estimating the transform for a model in canonical orientation makes the computation of the perpendicular distances straightforward. Since the points are transformed to match the unit square the distance to the closest side or closest corner of the square can be computed efficiently. Figure 2.18 shows the principle: the original points are mapped to the canonical square by the affine transform A^{-1}.

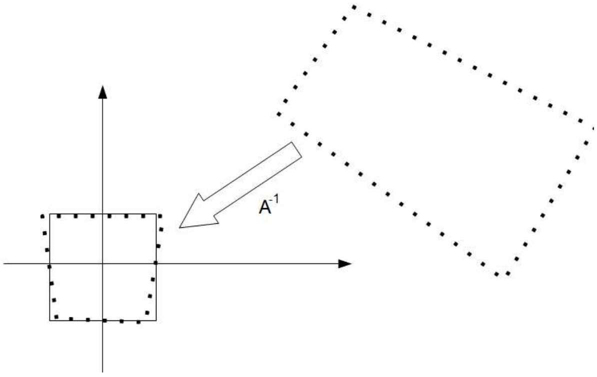


Figure 2.18: Illustration of the inverse affine mapping. The affine transform A^{-1} maps the point set (shown as dots) to canonical square. In this configuration the point square distance can be computed in an efficient manner.

The evaluation of the perpendicular distance of a 2D point $p = (x, y)$ to the unit square can be split up into three different cases (illustrated in figure 2.19):

1. The point is inside the unit square, in this case the perpendicular distance to the closest edge of the square is computed. This is simply the smallest absolute coordinate: $d = min(|x|, |y|)$.
2. If the point p lies outside the square but its projection is contained by one of the squares edges (these areas are marked by the 45^o hatching) the distance d is the absolute difference between the x-coordinate of the line and the x-coordinate of the point for vertical edges and the y-difference for horizontal edges.
3. For points that lie in the corner areas (shown in vertical hatching) the Euclidean distance to the corner point is reported.

In order to cope with outliers the Huber kernel is used to implement a robust distance function as proposed by Fitzgibbon [36]. In this work it was shown that the use of a robust error function can widen the basin of convergence significantly. The robust distance measure makes the estimation invariant against outlier points, at least as long as the initialization provides a sufficiently good starting position. When it comes to choose the Huber threshold the magnitude of the noise of the inlier points has to be known (at least approximately). Due to the fact that the distances are measured with respect to a canonical square, a fixed threshold can be chosen. In our experiments we set this threshold to be 0.1. That means that every point that is further than 10% of the square's side length is considered an outlier.

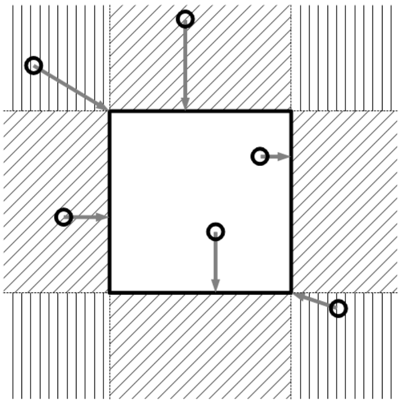


Figure 2.19: Illustration of the point to unit square distance. The three different cases are shown: The distance for a point inside the square is simply the absolute value of the smaller coordinate (due to the fact that the square is placed at the origin). For points outside the square whose projection is contained by one of the edges (these areas are marked by the 45^o hatching), the distance is also the absolute difference between the x-coordinate of the line and the x-coordinate of the point for vertical edges and the y-difference for horizontal edges. For points that lie in the corner areas (shown in vertical hatching) the Euclidean distance to the corner point is reported.

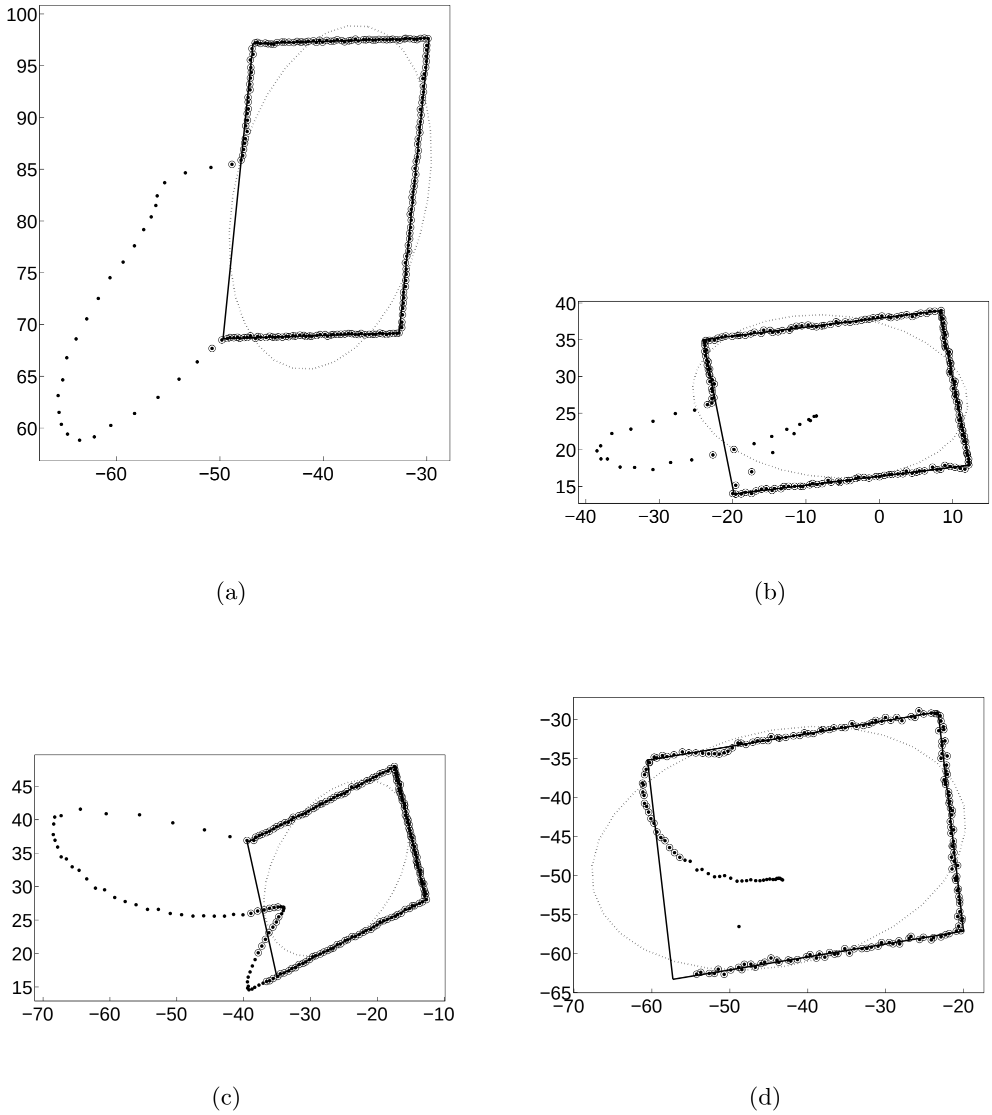
However, if the contour contains a significant amount of outliers (points that do not belong to the affine distorted square) the robust least squares optimization may still not converge to the correct solution. The main observation is that a proper initialization is still crucial for achieving robust convergence. Assuming that a sufficient amount of contour points is in fact inliers (at least 70%) a set of potential inlier points is determined by searching for the best fitting ellipse. This approach has the advantage that a fast RANSAC-based detection method can be used to provide an initial set of *good* points to start the non-linear square fitting. Figure 2.20 shows three examples of synthetic contour points and the best fitting ellipse is shown. The RANSAC approach robustly detected an initial set of square points (marked by circles around the points). For this subset of points an initial solution for the affine transform is computed and in a final optimization step all points are used to refine the solution. The computation of parameters of the forward transform A from A^{-1} is straightforward.

When the proposed method is used to detect squares and rectangles in perspective images the following limitation should be kept in mind:

- The estimated scale parameters model the scale differences that arise from the perspective distortion as well as the scale differences that are present in the physical rectangle. These two effects can not be separated without further constraints.

- The perspective projection maps a general rectangle (the square is a special case) to a convex quadrilateral. Therefore it is not possible to verify whether a primitive is a rectangle, a square or some general quadrilateral that is well described by the given transform. In order to do this, further geometric cues are necessary e.g. the 3D plane on which the primitives lie must be known.

Fitting affine squares to contours has many applications in urban modeling scenarios. The automatic detection of windows is one example. The examples shown in figures 2.20 and 2.21 illustrate the potential of the method.



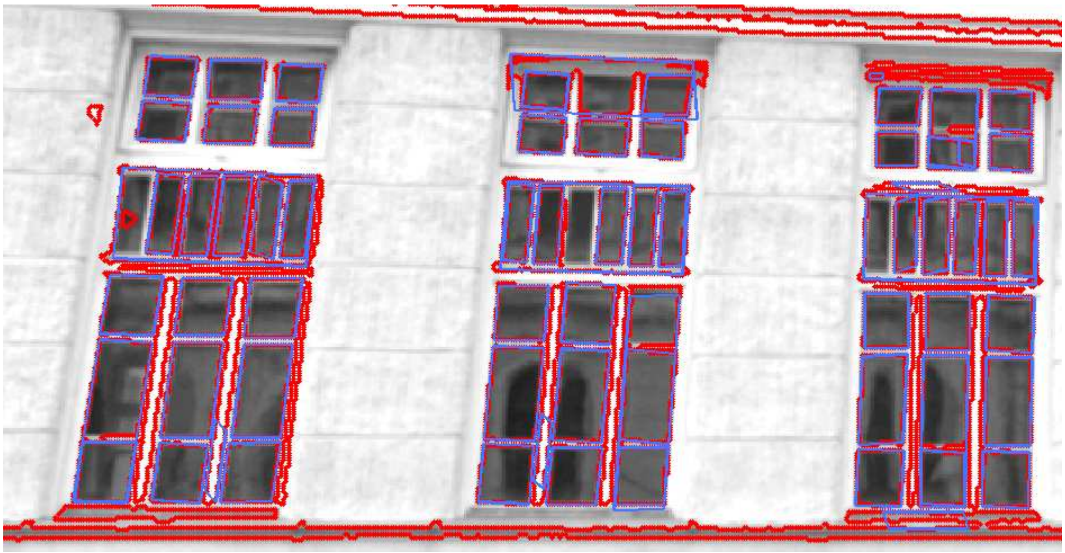
(a)

(b)

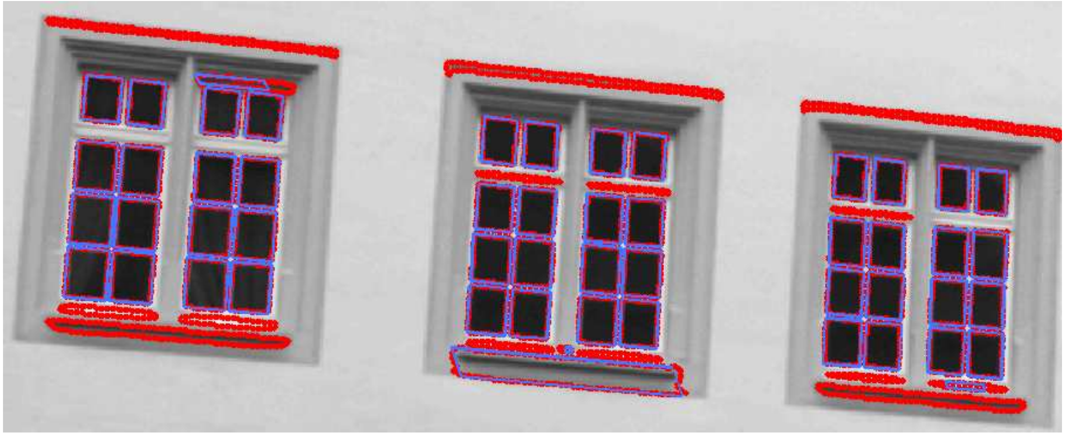
(c)

(d)

Figure 2.20: Examples for fitting affine distorted squares to noisy point sets. In each plot the points are drawn as solid dots, the detected ellipse as dotted line, the fitted affine square as solid line and the inlier points are marked by circles. Plot (a) shows the result for a point set with 15 percent outliers (30 points from 200), ($\sigma = 0.06$). Plot (b) shows the result for a point set with 15 percent outliers (30 points from 200), ($\sigma = 0.15$). Plot (c) shows the result for a point set with 30 percent outliers (60 points from 200), ($\sigma = 0.09$). Plot (d) shows the result for a point set with 30 percent outliers (60 points from 200), ($\sigma = 0.21$).



(a)



(b)

Figure 2.21: Examples for fitting affine distorted squares to contours of MSER regions. In each plot the contour points are drawn as red dots and the fitted affine squares as blue rectangles. Note that many squares are detected for the MSER regions of the window panes.

Chapter 3

Establishing point/feature correspondences

Contents

3.1 Introduction

This chapter addresses the problem of establishing point correspondences between pairs of images - the so called correspondence problem. Throughout the computer vision community the vision problem is recognized as one of the hardest problems and numerous publications propose solutions. What makes the correspondence problem hard to solve are strong changes in the visual apperance of a physical point mapped from different viewpoints. Another problem that occurs especially in scenes containing man-made structures is the presence of repeating patterns that result in large groups of nearly identical points. Traditionally the correspondences are established between point-like features. Descriptors are used to measure the similarity between a possibly corresponding point pair. In order to be discriminative descriptors should be invariant against viewpoint and illumination changes.

3.2 Descriptors

Descriptors are multidimensional vectors that serve as an abstraction for local or global properties of an image. Simple global descriptors are for example histograms, that represent one aspect of the image content. A gray value histogram e.g. is a vector where each bin corresponds to the frequency of occurrence of a specific gray value in the image. Whenever descriptors are used, a part of the original information is given up in order to gain invariance. Histograms are useful to compute statistics on the pixel values but bear no longer information of the location of a particular pixel. Scale invariant features, as the name implies, sacrifice the information about the features scale and location to gain an invariant representation.

A hierarchical classification of descriptors can be done by regarding the levels of invariance:

- rotational invariance
- rotational and scale invariance
- affine invariance

The most simple descriptor is an axis aligned window containing the image content around a point. This simple descriptor is only invariant against translational shifts and breaks down quickly in the presence of perspective transforms or even small rotations.

Commonly used features are the affine invariant ones, since perspective transforms, as they occur in wide baseline setups can be locally approximated by an affine transform. Typically an interest point detector provides locations at which a local affine invariant descriptor is computed. Based on the assumption, that the area around the interest point is planar or sufficiently smooth, an affine invariant descriptor is useful. Several methods have been proposed in literature e.g. by Baumberg [14], Lowe[77], Schmid and Mohr [110]. Mikolajczyk and Schmid [86] evaluated the performance of several local descriptors. The most challenging problem in these approaches is to find the correct scale i.e. the spatial extension of the support region around the point. Other methods define an invariant region by finding a stable border as proposed by Schaffalitzky and Zisserman [109], Tuytelaars and Van Gool [127] or Matas et.al [80]. Larger regions seem to be preferable because they allow a more distinctive description, but on the other hand are more likely to contain occlusions if the same region is viewed from a different viewpoint. Larger regions may also deviate from the planar case or exhibit large perspective distortion. The computation of descriptors often involves a transform of the image content around a specific point to a different representation. In case of SIFT features [78] the area around a point is transformed into a set of orientation histograms, another example is the use of generalized moments as proposed in [127].

3.3 Robust Image Correspondences using Invariant Features

This section introduces a novel method for computing affine invariant features using Zwickels*. The proposed hat is especially well suited for images of man-made structures. Zwickels are sections defined by two intersecting line segments, dividing the neighborhood around the intersection point into two sectors. The information inside the smaller sector is used to compute an affine invariant representation. We rectify the sector using the line information and compute a histogram of the edge orientations as a description vector. The descriptor combines the advantage of accurate point localization through line intersection as well as high ability to discriminate through use of a larger image region compared to descriptors computed around the points. A geometrically motivated approach for selecting the characteristic scale of Zwickels is used for delineation. Compared to other affine invariant descriptors we demonstrate that our method avoids the problem of depth discontinuities. In several matching experiments we show that our features are insensitive against viewpoint changes as well as illumination changes.

A Zwickel is formed by the intersection of two lines, where the intersection points of the line segments serve as interest points. The principal idea behind this approach is, that the area between intersecting lines is in many cases planar. Unlike other methods that compute the descriptor for a symmetric or skew-symmetric region around the interest point, we use the dividing property of the line segments to compute the descriptor only for the smaller sector. This has the advantage, that if two sectors match, we compare only the correct parts and thereby achieve a higher discrimination ability, especially if lines are lying on depth discontinuities. Our approach is split up into two steps: first we detect potential Zwickels by searching for intersecting line pairs. This step yields accurate points of interest and subdivides the region around this point into two sectors. The lines therefore automatically provide a segmentation by dividing the region around the interest point into two sectors.

[*] German: *zwicken* : to nip

In the second step we compute affine invariant descriptors for those sectors that are enclosed by the intersecting lines. The computation of the affine invariant descriptor involves a rectification of the enclosed sector and the construction of a histogram of the edge orientations. It is clear, that the proposed interest points can only be detected in images, where a sufficient number of lines is present - this is true for images containing typical man-made structures. The geometric accuracy of the intersection points is higher than those of corner-based points of interest.

The outline of subsequent sections is as follows: In section 3.4 we describe the detection of Zwickels and the computation of the affine invariant descriptor. Section 3.4.5 shows the application of the Zwickel descriptors for image matching. Experiments with real and synthetic images are presented in section 3.4.7 and we close with concluding remarks.

3.4 Zwickel detection and description

In the following we describe how Zwickels are detected, explain the rectification process in more detail and address the computation of the affine invariant descriptor.

3.4.1 Zwickel detection

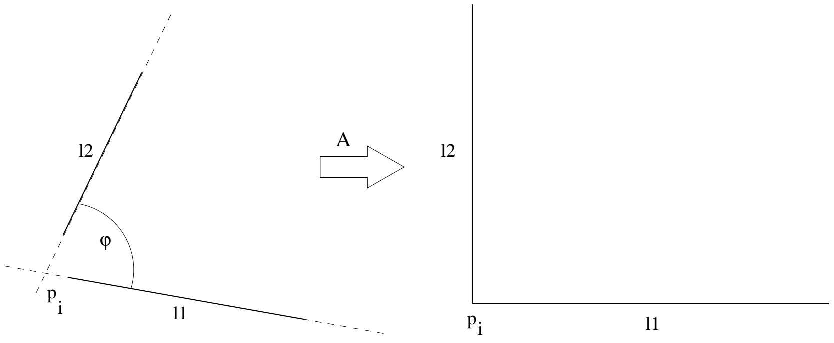
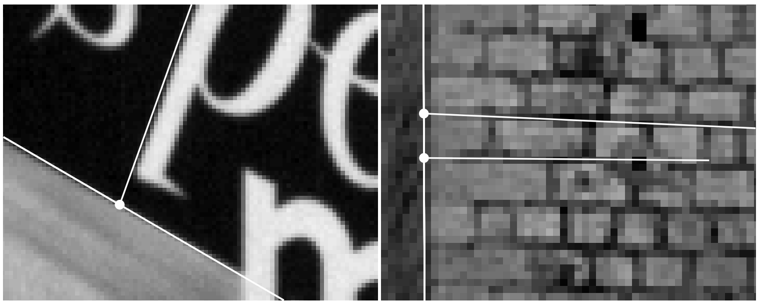


Figure 3.1: Left: geometry of a Zwickel: p_i is the intersection point of the lines l_1 and l_2 which are extended by a factor (extensions are shown dashed) to ensure intersection. Right: For the rectification the lines l_1 and l_2 with the enclosed angle φ are mapped to an orthogonal frame using the affine transform matrix A. The transform maps the intersection point p_i to origin and the lines to the axes of the coordinate system.

The detection of Zwickels is performed as follows: In the first step 2D line segments are extracted from the image, those segments are extended by a predefined factor to ensure that lines, that are close enough, will intersect. All reported intersections are then handed over to the Zwickel formation procedure. For the line detection we use a hierarchical approach, that finds straight lines in a coarse-to-fine pyramid search. In every pyramid layer we extract Canny edges [24] with sub-pixel accuracy and fit straight line segments to sets of collinear edges. In order to compute intersections we extend the resulting line segments to ensure a sufficient number of line intersections. The detection of Zwickels is affine invariant. The lines of the detected Zwickels are ordered clockwise to ensure the correct correspondence between the lines of two matching Zwickels. As already mentioned we extend the originally extracted lines, therefore the intersection points may lie in a homogeneous region. This is one of the additional advantages over point-of-interest methods that rely on detection of location of high gradient curvature such as the Harris corner detector [49]. Figure 3.2 shows two examples of extracted Zwickels with low gradient curvature at the intersection point.



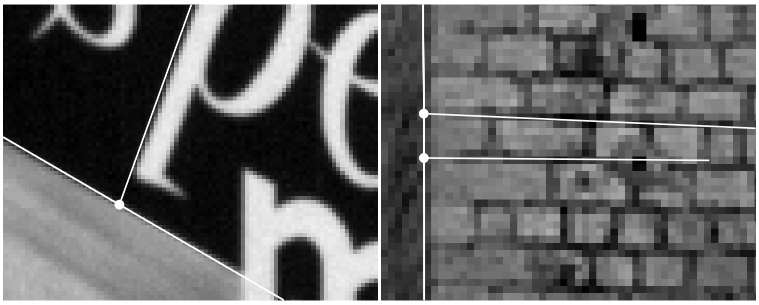
(a) (b)

Figure 3.2: Examples of extracted Zwickels where the intersection point (denoted by the circle) of the two extended lines does not lie on a location of high gradient curvature i.e. no Harris corners would be detected at the intersection point.

3.4.2 Detection of the characteristic scale

Since by definition the Zwickels are created by intersecting two line segments they do not provide information about their characteristic scale. However scale information is necessary do compute an affine invariant descriptor. In contrast to local descriptors that perform the scale selection using a signal theoretic approach (e.g. DOG, Hessian) the often complex visual appearance of the region inside a sector bounded by the Zwickels lines often makes this approach futile.

Figure 3.3 shows examples of Zwickels detected for typical facade images where the sector between the bounding lines contains complex patterns.



(a) (b)

Figure 3.3: Examples of extracted Zwickels where the intersection point (denoted by the circle) of the two extended lines does not lie on a location of high gradient curvature i.e. no Harris corners would be detected at the intersection point.

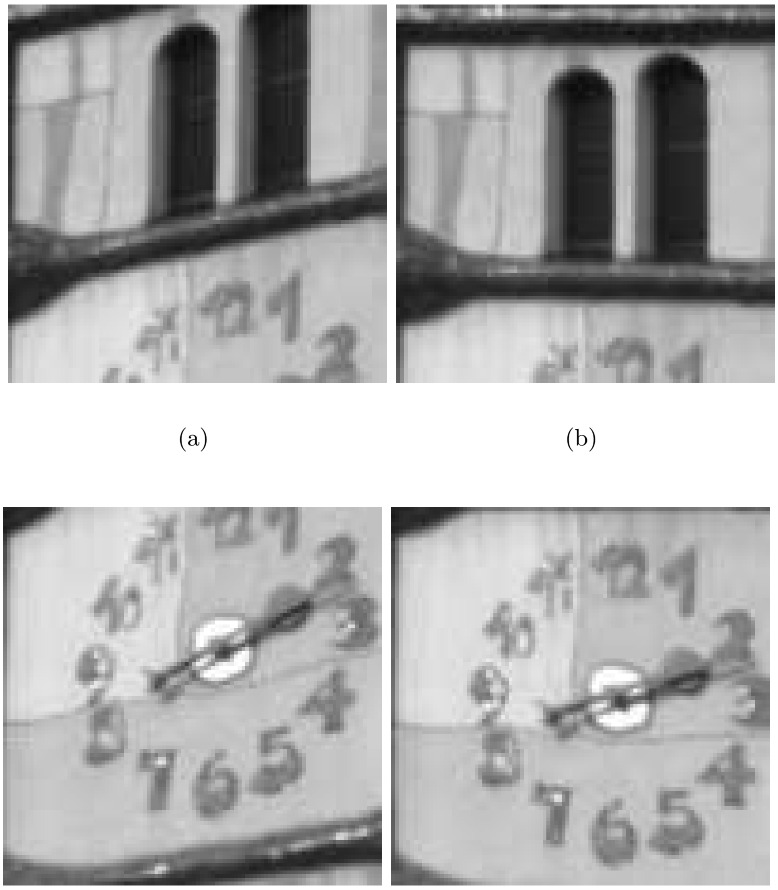
Man-made structures, especially buildings comprise many rectangles - nearly every window, door sign etc. is of rectangular shape. This rectangular structures can be represented by a combination of Zwickels simply by combining two edges that share a corner point. This observation leads to a geometric approach for Zwickel delineation.

3.4.3 Zwickel rectification

In order to compute an affine invariant representation of a Zwickel, we map the image data inside the sector that is bounded by the lines to an orthogonal frame (see Figure 3.1). An affine transform is computed from one corresponding point (the intersection point is mapped to the origin) and the two line directions. The image region in the sector is then rectified by applying the affine transform that maps the sector to an orthogonal frame with the intersection point as origin and the lines as axes of the coordinate system. Equation 3.1 shows the general form of an affine transform and its decomposition into a rotation, scaling and shear transform. The rectification eliminates four of the six unknowns of the affine transform: the translation $[t_x, t_y]$ through shifting the intersection point to the origin and rotation φ and skew s through mapping the lines as orthogonal axis. The remaining unknowns are the scale factors s_x and s_y. In order to determine the unknown scale we use a similar approach as in [77, 82]. Both approaches use a scale space search to find the correct scale of the support region.

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \\ &= \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\varphi) & sin(\varphi) \\ -sin(\varphi) & cos(\varphi) \end{bmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \end{aligned} \tag{3.1}$$

Figure 3.4 shows two examples of the rectification step.



(a)

(b)

(c)

(d)

Figure 3.4: Orthogonal rectification: (a) and (c) original image regions inside Zwickel. (b) and (d) rectified image regions.

3.4.4 Descriptor

In order to achieve affine invariance we apply a scale invariant descriptor. The descriptor is inspired by Lowe's [77] SIFT-features.

We first calculate the edge orientation φ and magnitude m at each pixel inside the rectified frame I:

$$m(x, y) = \sqrt{(I_{x-1,y} + I_{x+1,y})^2 + (I_{x-1,y} + I_{x+1,y})^2} \tag{3.2}$$

$$\varphi(x, y) = atan((I_{x-1,y} + I_{x+1,y})/(I_{x-1,y} + I_{x+1,y})) \tag{3.3}$$

An orientation histogram is used as a region descriptor, the magnitude and the distance of the pixels from the origin are used as a weight. More formally the histogram is calculated as

$$H(\theta) = \sum_{\varphi \epsilon \mathcal{N}} \delta(\theta, \varphi) * w_{\varphi}, \tag{3.4}$$

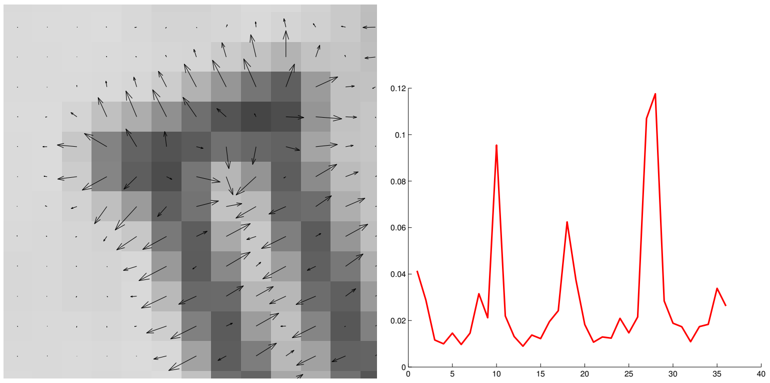
where $H(\theta)$ is the value for bin θ ($\theta \in [0°, 1° \ldots 360°]$) and φ denotes angle values in a neighborhood $\mathcal{N}$ inside the Zwickel, w_{φ} is the weight of φ and $\delta(\theta, \varphi)$ is the Kronecker delta function. The angles φ are quantized in accordance with the histogram bins θ. The weight w_{φ} is computed from the magnitude of φ and a function decreasing with increasing radius r from the origin (x_0, y_0). We use a Gaussian function thus $w_{\varphi}(x, y) = m(x, y) * g(r)$, with $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ and $g(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-r^2}{2\sigma^2}}$

The parameter σ of the Gaussian function has to be adapted according to the detected scale. Due to the use of image derivatives illumination insensitivity is also achieved.

3.4.5 Matching

In the matching step we want to detect similar regions in an image pair. Using the Zwickel representation it is easy to implement several pre-selection criteria to speed up the matching by reducing the number of putative candidates. The pre-selection is performed on the basis of geometric constraints as well as on image information. We only allow a maximal angle difference between corresponding lines of a Zwickel candidate pair. Furthermore we enforce the lines to have the same gradient direction. If a Zwickel encloses a darker region than the surrounding, the two lines have different gradient directions and therefore different line types.

Other pre-selection criteria for candidates e.g. by comparing the difference of the gray-value median for the Zwickels can be easily implemented. For the remaining candidates



(a) (b)

Figure 3.5: Visualization of orientations in the rectified frame: (a) image region with vectors visualizing the edge orientation (vector length corresponds to the magnitude). (b) histogram of edge orientations

we detect the most similar ones by comparing the descriptors. In order to accomplish this task we have to choose a proper distance function for the comparison of the orientation histograms.

3.4.6 Distance functions

Since the descriptors described in section 3.5 are histograms we use probabilistic distance measures to describe the similarity. Distance measures for histogram comparison are the L_1 and L_2 norm, the Bhattacharyya distance, and the Matusita distance. The earth movers distance is a more complex method for histogram comparison and is computed by solving the so called transportation problem, proposed for image indexing by Rubner et.al [107]. Huet and Hancock [56] give a comparison of the performance of this measures for histogram comparison. Following the conclusions of Rubner we chose the Bhattacharyya distance which is defined as:

$$D_{Bhatt}(H_A, H_B) = -ln \sum_i \sqrt{H_A(i) \cdot H_A(i)} \tag{3.5}$$

The Zwickel pair with the smallest distance is the most similar in terms of the histogram comparison.

3.4.7 Experiments

We carried out several experiments to show the performance of the proposed method. In all experiments the region size was 30 × 30 pixel. In order to increase the robustness of the matching we also compute the normalized correlation coefficient cc for the rectified image patches. The distance function therefore modifies to:
$D = D_{Bhatt}(H_A, H_B) * (1 - cc(A, B))$ where A and B denote the two rectified image patches and H_A and H_B are the orientation histograms for the image patches. In the first experiment we assess the invariance of the descriptor against viewpoint changes. Sequences of several box-like objects were acquired by a turntable setup. The rotation between two subsequent images is five degrees resulting in a 72 image series. A key image is selected and we perform the matching with all subsequent images. For evaluation purposes we keep thirty percent of the best matches (smallest D) and determined the number of correct matches by calculating the epipolar geometry. Figure 3.7(a) and Figure 3.7(b) show the rate of correct matches versus the rotation angle between the camera of the key image and the camera of the second image used for the matching. The correct matches are the inliers resulting from computing the essential matrix. The experiment is carried out with two different versions for the support region: Version one uses the sector as described in our approach. In version two the support region is centered skew symmetric around the point of interest. This comparison assesses the increase in discrimination ability when using only one sector of the interest point's surrounding. The inlier rate for our approach is represented by a solid line, the dashed line is the inlier rate for the skew symmetric support region.

Figure 3.6 shows the differences in the used support region.

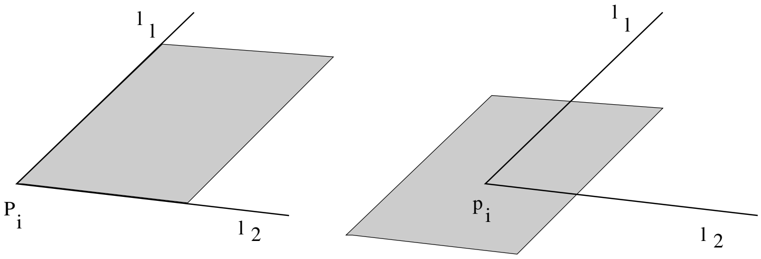


Figure 3.6: Illustration of the two cases for the support region. Left: support region lies inside the sector defined by the intersecting lines. Right: support region lies skew-symmetrically around intersection point

Figure 3.7(a) shows the results for the turntable sequence for real images. One can clearly see the superior behavior of the sector representation (approx. 20 percent increase in performance). The variance can be explained by occlusion effects e.g. when a new face of the box appears and the number of possible candidates increases or when a face disappears and the number of candidates drops. Our approach outperforms the version with the skew symmetric support region as the rotation between the cameras increases. In Figure 3.7(b) illustrates the results for the synthetic turntable sequence. The scene consists of a planar object with several differently structured textures 'glued' on it. Due to the lack of depth discontinuities the performance between the two versions for the support region differs less, which again nicely demonstrates the superiority of Zwickels on depth discontinuities.

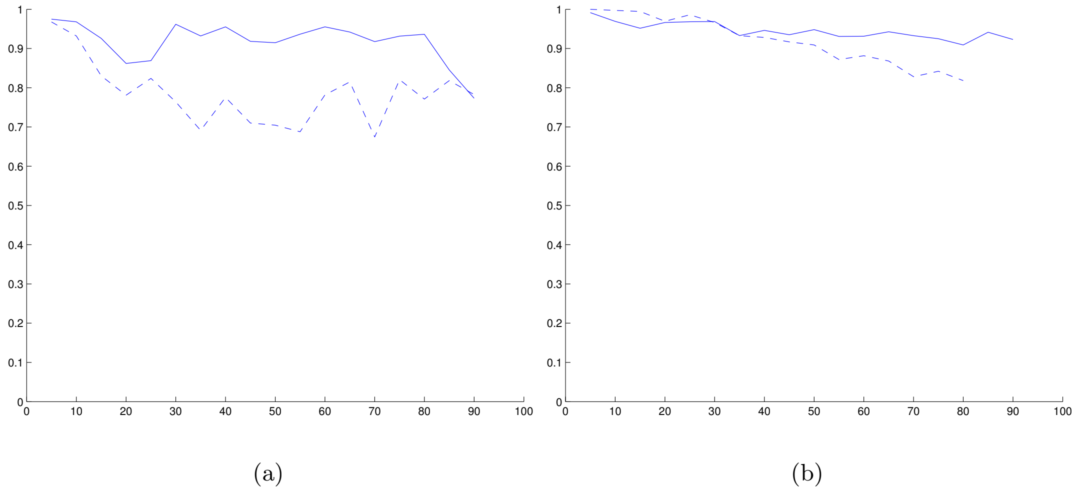


Figure 3.7: Illustration of the invariance against viewpoint changes: The rotation between the two cameras is increased in five degree steps from five to ninety degrees. The continuous line is the result for our approach, the dashed lines is for the centered support region. (a) shows the results for the data of the turntable sequence for real images. The variance results from occlusion effects, when new faces appear or other vanish. (b) illustrates the results for the synthetic turntable sequence.

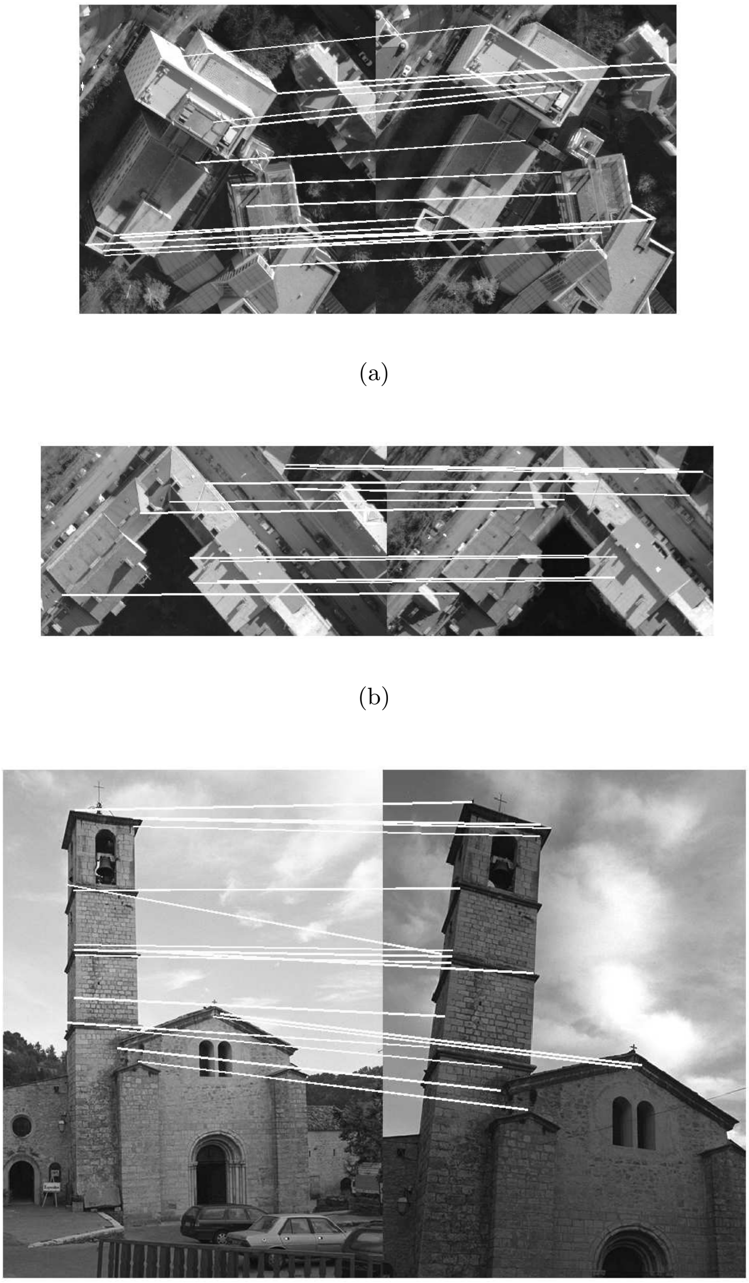
In the following experiment we took several image pairs and evaluated the matching performance. Figure 3.8 shows the 30 percent best matching correspondences for those image pairs. Table 3.1 lists the results for four different image pairs. Results using other images are similar. In column 2 we list the number of total matches found, column 3 shows the number of best matching correspondences used for estimating the epipolar geometry. In column 3 and 4 we list the number of inliers and outliers accepted or rejected by enforcing epipolar consistency. Note that all image pairs show a significant rotation between views. It is clearly seen that our novel method produces many good matches

Object	total matches	matches used for epipolar geometry	inliers	outliers
aerial image pair 1	67	67	59	8
aerial image pair 2	51	51	42	9
turntable images 'Obi'	229	68	66	2
virtual turntable images	282	84	80	4
Valbonne image pair	112	50	41	9

Table 3.1: Evaluation of the matching performance. Results are given for five image pairs. Note that for the turntable images as well as for the virtual turn table scene most of the inliers lie inside a planar region, for the aerial image pairs several matches lie on depth-discontinuities where the Zwickel-based descriptor is well suited. For the Valbonne image pair several matches were found at depth discontinuities since many prominent lines were found on the borders of planar regions.

and only few outliers. The matching, including the estimation of epipolar geometry, takes between 6 and 14 seconds on a Pentium 4 machine with 2.4 GHz.

Our experiments show, that these descriptors are invariant against viewpoint changes as well as illumination changes. Our method is suitable for images where a sufficient number of lines and therefore Zwickels can be extracted and the sectors inside the Zwickels provide enough texture information to distinguish competing candidates. A further possible improvement is the use of a more powerful distance measure for histogram comparison, such as the earth-movers distance.



(a)

(b)

(c)

Figure 3.8: Matching results for two aerial image pairs and a terrestrial image pair (Valbonne church). For clarity only 30 percent of the best correspondences are shown. (a) Image pair 1. (b) Image pair 2. (c) Valbonne image pair

3.5 Global Descriptors for robust correspondence detection

3.5.1 Introduction

In city modeling applications typical scenes consist of dominant facades, vegetation, non-static objects like pedestrians, moving and parking cars etc. The camera pose can only be estimated from point correspondences on static objects. Matches between non-static objects are removed during the image orientation step. Matches on cars are often geometrically unstable due to smooth or specular surfaces. This facts imply that it is desirable to establish matches on the rigid facade planes, however the appearance of building facades makes a robust matching difficult. The main reason for this is the presence of repeating patterns. Repeating patterns are common in man-made structures and make robust correspondence estimation based on local descriptors a challenging task. Although there has been substantial progress in this area by methods like [61, 78, 83–85] the robust estimation of correspondences under the presence of repeating patterns is still challenging.

The reason for this difficulty is that local features, as the name implies, use only a relatively small portion of the image in their immediate neighborhood to compute a robust descriptor. The robustness is achieved when the region around the point is planar - in this case an affine invariant representation of the patch is possible. This strategy makes local desriptors well suited for wide baseline matching but introduces ambiguities when the same pattern occurs repeatedly.

Figure 3.9 illustrates the problem: it is not clear which of the window corners in the right image corresponds to the window corner in the left image, especially when only the local neighborhood of the corner is considered.

A solution to this problem is the incorporation of a larger area for the derived descriptor. However, the larger the area becomes, the higher is the probability that the enclosed region is no longer planar. In this case the computation of an affine invariant descriptor is no longer possible. Even if the region is planar an affine invariant representation is not sufficient to describe the patch, since for larger regions the perspective distortion becomes dominant.

In the light of this findings several methods that derive descriptors for points-of-interest, incorporating information from the whole image or a large portion of the image, have been introduced. Mortensen et al. [89] combines local descriptors and shape context descriptors. Shape context [17] uses the distribution of points to compute a simple descriptor for a point-of-interest. While this method is originally intended for object recognition

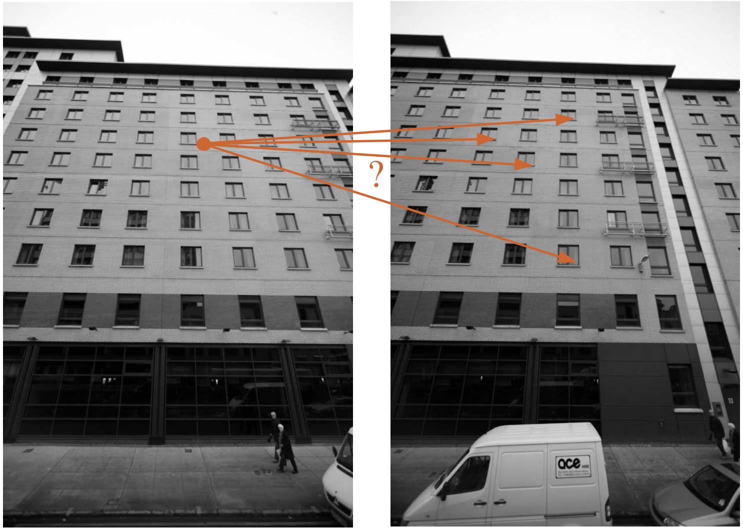


Figure 3.9: Establishing correspondences between views that contain repeating patterns is a challenging task. The window corner marked by the circle in the left image matches with a great number of window corners in the right image.

the combination with a local descriptor results in a more discriminant descriptor.

Tell and Carlsson [123] proposed a method that incorporates the topology of neighboring points via intensity profiles to improve the stability of the matching process. The method is based comparing gray level profiles that are sampled between corner points. If these profiles lie on a planar surface, a homography relates the image contents between different views of the same surface patch. Furthermore for every corner on the patch the cyclic ordering to all other corners on the patch is preserved under perspective and therefore constitutes a powerful invariant. For the N nearest points of a corner in an image the descriptor is formed by computing the grey level profiles from the center point to its surrounding neighbour points. The topology constraint is then enforced on the configuration of profiles emitting from one corner point.

In this section we present an overview of the shape context descriptor and show its usefulness for image matching as proposed by Mortensen. Furthermore an introduction to the template matching method based on geometric blur by Berg and Malik [18] is given. We then present a method for approximating the geometric blur operator for computing semi-global and global descriptors and compare them with the approach of Mortensen et al. [89]. These new descriptors allow for a robust estimation of correspondences between image pairs in the presence of repeating patterns. We test the performance of the various global descriptors for correspondence estimation and image orientation in the field of city modeling.

3.5.2 Combining SIFT and Shape context

The approach of Mortensen et al. is motivated by the search for consistent point correspondences in the presence of repeating patterns. The robust SIFT descriptor provides scale invariant local information for a point and the shape context descriptor is used to incorporate information of a larger surrounding area. Shape context [17] descriptors are histograms where the entries are composed of the number of edge points that fall into individual log-polar bins.

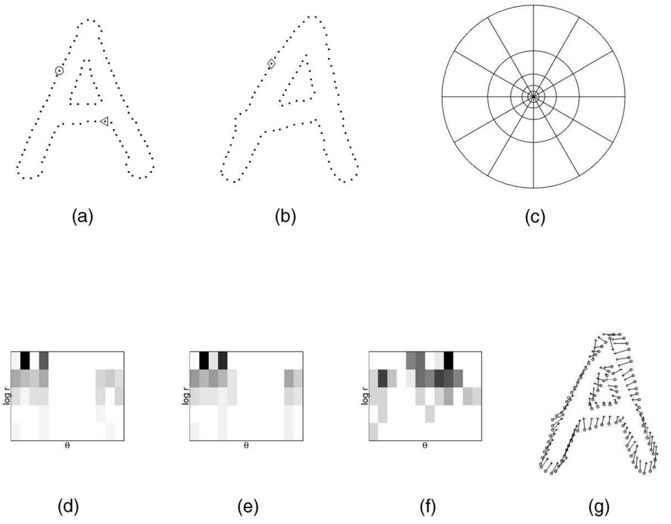


Figure 3.10: Principle of the shape context descriptor for contour points: (a) and (b) are the contour points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts (five bins for $log(r)$ and 12 bins for $theta$). In (d), (e) and (f) example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ in (a) and (b) are shown. Each shape context is a log-polar histogram of the coordinates of the point set. The radius r is measured using the reference point as the origin. (Dark=large value). In (g) correspondences found by bipartite matching, with costs defined by distance between histograms, are shown. Figure taken from [17]

Figure 3.10 illustrates the approach for the contour points of two instances of the letter A. The value of each bin in the log-polar histogram is the number of edge points that fall into the corresponding sector. The exponentially increasing area of the histogram bins makes the descriptor less sensitive to perspective distortions and scale changes. In [17] the points used were silhouette points of 2D shapes, whereas Mortensen et al. used the maximum curvature at each pixel. For a given pixel at position (x, y), the maximum curvature is the absolute maximum eigenvalue of the Hessian matrix. This makes entries invariant against changes in image contrast. The diameter for computing the descriptor is equal to the diagonal of the image. The shape context descriptor is inherently not invariant against rotation, that means if a shape is slightly rotated, the shape context histogram changes in the way that the entries are shifted to neighboring bins and the distance between the histogram of a shape to its rotated version increases. A way to overcome this, is to make use of the key orientation provided by the SIFT descriptor and to compute all angles w.r.t. this key orientation. This strategy makes the shape context descriptor invariant against rotation.

In the matching stage the similarity measure for comparing two descriptors is the weighted sum of the Euclidean distance of the SIFT descriptors and the χ^2 distance of the global shape context histogram.

The experiments demonstrated that a significant improvement of the matching rate can be achieved.

A disadvantage of the proposed method is that the computational overhead for generating the global descriptor is high. This is caused by the necessity to perform a sampling on a large patch in order to fill the bins of the shape context histogram. This overhead makes the approach cumbersome for correspondence estimation in high resolution images.

A strategy to overcome the exhaustive sampling for the global descriptor will be presented in the following.

3.5.3 Approximated shape context

The motivation for this new semi global descriptor is to achieve a computationally efficient description of a relatively large region around a point of interest. The proposed descriptor is inspired by both shape context[17] and the concept of using geometric blur for template matching presented by Berg and Malik [18]. Figure 3.11 shows the principal concept of geometric blur:

The general design of geometric blur is based on the observation that in case of template matching the positional uncertainty inside template windows is zero at the central pixel of the window and increases towards more peripheral positions. Figure 3.12 illustrates this for two template windows of a facade detail: With increasing distance from the window center the number of mismatched pixels and therefore the difference increases.

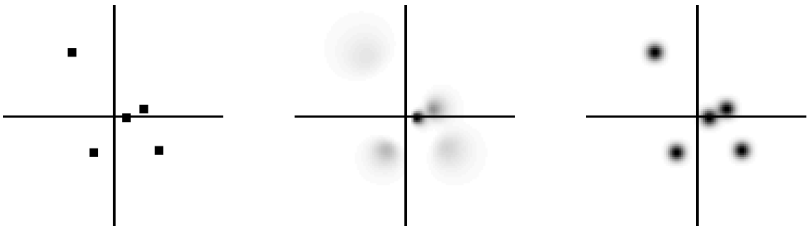


Figure 3.11: Principle of geometric blur: The leftmost figure depicts a 2D signal; The figure in the middle shows the effect of geometric blur - in this case the amount of blurring depends on the distance from the origin and the rightmost figure shows the result of Gaussian blur with an uniform kernel

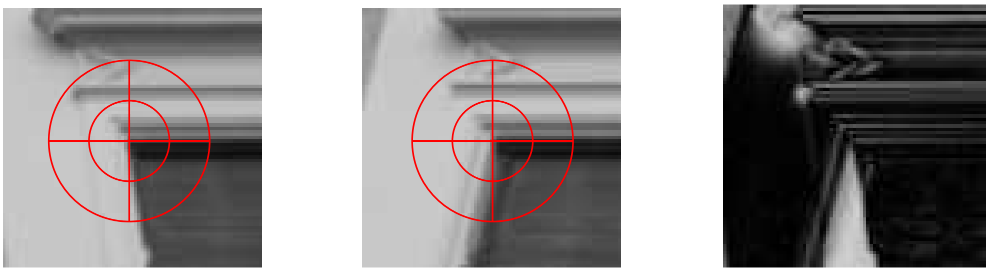


Figure 3.12: Left and middle: Two template windows of 81 × 81 pixels. Right: Difference image.

Berg and Malik formally defined the geometric blur operator as follows:

$$G_I(x) = \int_y I(x-y)K_x(y)dy \tag{3.6}$$

Where $I(x-y)$ is the image centered at position x and $K_x(y)$ is a spatially varying kernel.

In oder to compute descriptors for similarity measurements between points of interest the geometric blur operator is applied to oriented edge filter responses computed on an image region (template) around a point. The original approach uses six different orientation filters resulting in six 'channels'. The blurred templates for all channels are then compared by adding together normalized cross correlation using geometric blur from each of the channels. The authors performed tests using their method for object recognition and wide baseline matching. Computational costs are also an issue in this approach, especially the split of an image patch into several channels and performing the geometric blur on each channel separately is expensive.

In our new approach, the concept of increasing blur from the point of interest is realized by stacking layers of exponentially growing bins. Thus the descriptor is basically a hierarchically sampled image patch. Figure 3.13 illustrates the approach: The sampled area increases exponentially with increasing distance from the center. This strategy preserves the local information close to the center and generalizes at larger distances. One of the key advantages of this descriptor is the low computational cost. By using integral images the cost for computing the descriptor is linear in the number of layers since the mean grey value of a rectangular, axis aligned patch can be computed in linear time from four sampled values in the integral image. Another advantage is that the size of the descriptor is linear in the number of layers, while the covered area increases exponentially.

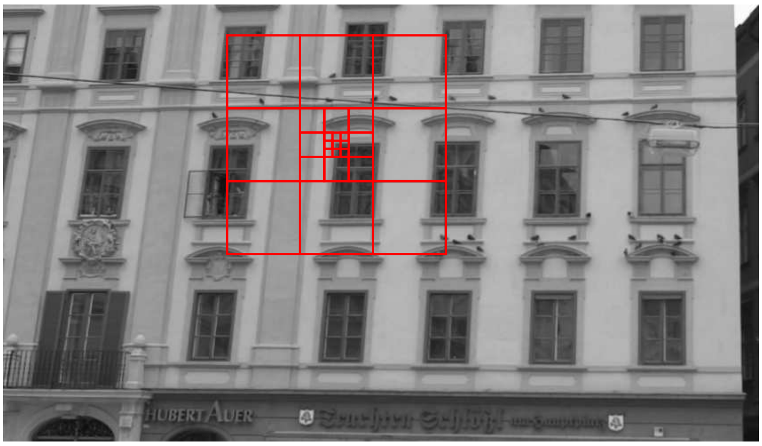


Figure 3.13: Semi global descriptor principle. The red grid depicts the sampling region covered by exponentially increasing sampling areas.

In order to make the descriptor rotation invariant we assign a key orientation to each corner point as described in subsection [89] and sample the descriptor entries at the corrected positions (see Figure 3.14). The sampled regions are still axis aligned to retain the efficiency of the fast mean computation with the integral image approach.

In our implementation the innermost area consists of nine samples at pixel resolution resulting in an initial box size of three for the first layer that is computed from the integral image. The side length of sample regions then increases by powers of three (3, 9, 27, etc.), so that for a number of four layers the diameter of the covered region is 81 pixel.

In contrast to the exponentially increasing area covered by the descriptor the dimension of the descriptor grows linearly in steps of eight for each new layer. A descriptor with four layers has a dimension of 33.

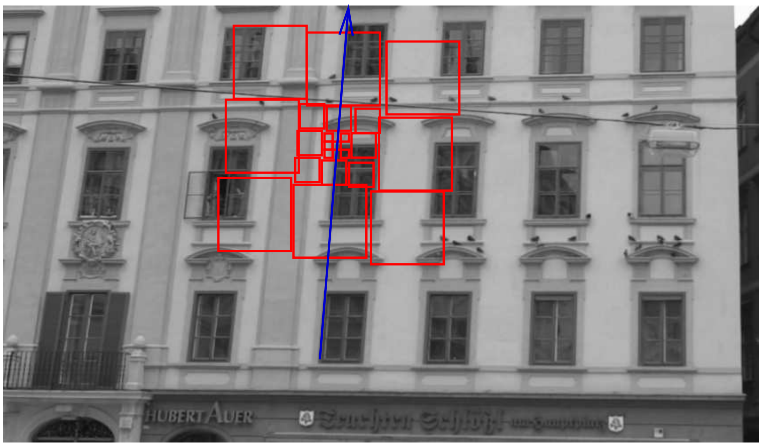


Figure 3.14: Rotation invariant semi global descriptor. The axis aligned samples are shown as red squares and the key orientation for the corner point is shown in blue.

3.5.3.1 Relation between shape context and geometric blur

The proposed approximation scheme can be used for the efficient computation of both, shape context and geometric blur descriptors. When using sparse input images the only difference between shape context and geometric blur is that for computing the entries into the geometric blur descriptor, the value for a particular box resulting from the integral image access is normalized and therefore equivalent to a box shaped mean filtering. In case of shape context the sum over all pixels in the box is not normalized by the box area and represents therefore a measure for the frequency of occurrence of features.

The approximation accuracy for shape context is higher, because the boxes constitute a good approximation of the log-polar bins. Approximation for geometric blur is less accurate since the level of blur jumps significantly with every layer and instead of a Gaussian convolution kernel, a box filter is used.

The similarity measure for the matching step is the normalized cross correlation for the geometric blur and the χ^2 distance for the shape context histograms.

3.5.3.2 Global or semi-global?

Since the origin of shape context is in the field of object recognition the computation of the descriptor always includes all available information. This is justified by the fact that a segmentation for the objects is available in the learning phase: Either the outlines of objects were already available as point sets or the image capturing was performed under controlled lighting and with a well defined background. In the case of correspondence estimation between image pairs the situation is completely different: The goal is to improve the discrimination ability by augmenting local descriptors through incorporation of image information of a larger area around the point-of-interest. As Mortensen et al. mention in the conclusion of their paper the computation of shape context descriptors for the whole image makes it necessary to ignore bins that lie outside the image bounds and for bins that lie partially inside the image a proper normalization is necessary.

For the image matching task the descriptor radius for each point-of-interest is determined by the number of layers used in the integral image approximation of shape context. Regardless of the radius it is required that the descriptor's footprint is fully inside the image. For practical reasons the number of layers is restricted to four or five (resulting in radii of 40 and 121). This restriction makes the proposed descriptor semi-global.

In the experimental section a comparison of shape context versus the proposed approximation scheme will be carried out and it will be shown that the normalization of the histogram entries by the box area results in a descriptor that is similar to geometric blur and has a higher discrimination ability. Experiments performed on typical facade data indicate that the developed descriptor is invariant against rotation and shows good performance under scale changes and perspective distortions.

3.5.3.3 Advantages of the approximation

An important advantage of the approximation is the reduced computational effort for computing the descriptors. The original method for computing shape context works on point-like features and the complexity is therefore linear in the number of points. A solution to speed up the procedure for large point sets is to render the points into an image, derive the integral image from and perform the approximation. This strategy reduces the complexity to $O(8n)$, where n is the number of layers.

Compared to the shape context descriptor used in [89], where the content of each bin is directly sampled from an image, the gain is even higher, since the number of image samples that are necessary to fill the bins grows exponentially.

Chapter 4

Feature-Based Modeling

The following two sections are concerned with the extraction of 3D primitives from images. In contrast to popular methods that compute dense depth maps from two or more input images the method described here use image features as input data. More concretely the features used here are edgels and primitive derived from edgels (contour chains and straight line segments) that are extracted with sub-pixel accuracy in a preprocessing step. Both methods have in common that so called sweeping strategies are used to traverse the 3D search space. Due to the use of efficient geometric data structures the methods are fast, and robust similarity metrics allow for the generation of reliable 3D hypothesis.

The first method is concerned with the extraction of 3D plane hypothesis from two or more images with known relative orientation. The particular scenario for this method is the detection and delineation of facade planes in a terrestrial city mapping setup.

Contents

4.1 Plane parameter estimation by edge sweeping

In this section a new method for the automatic generation of 3D plane hypothesis based on a sweeping process is presented. The method is feature-based in contradiction to earlier methods which rely on image correlation. The features used are edgel elements ('edgels') which are extracted in a preprocessing step. The experiments show an improvement in speed as well as in accuracy.

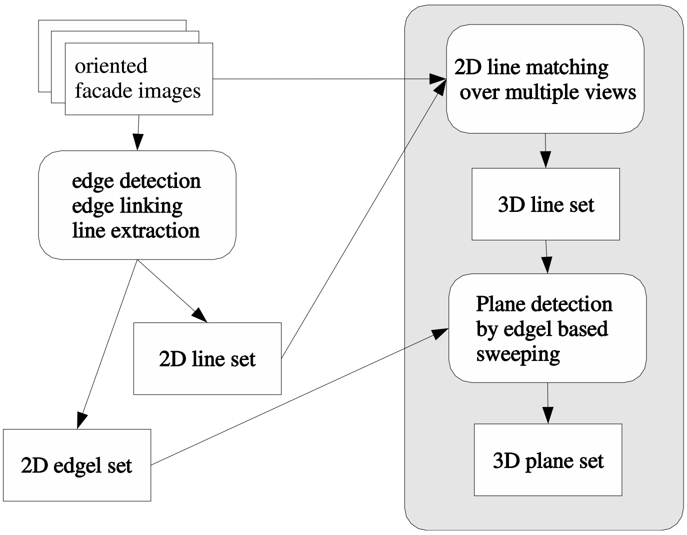


Figure 4.1: Sketch of the work flow: On the left side the feature extraction work flow is visualized, on the right side the 3D modeling pipeline is depicted. The grey shaded areas are discussed in this paper.

4.1.1 Introduction

The automatic generation of 3D models from digital images is a popular topic. Especially in the field of city modeling, where huge amounts of data have to be processed, a high degree of automation is desirable. The work-flow of our system is shown in Figure 4.1. In this paper we discuss the gray shaded parts, which describe a feature-based algorithm for 3D plane hypothesis generation based on a sweeping process. A set of images with known exterior orientation together with extracted 2D lines are the input data. A line matching method is applied to generate 3D line hypothesis and subsequent plane-based homographies for image pairs. Using the homographies the edgel sets in the vicinity of the projected 3D lines are swept in order to detect the optimal orientation of the half planes. The sweeping process is also used to validate the 3D line hypothesis. Our approach is based on image features, namely edgels, instead of using plain image information as in earlier approaches by Zissermann et al. [4] and Baillard et al. [5]. The method is also comparable to the technique described by Coorg and Teller in [29]. The advantage is a significant speedup and better geometrical accuracy. Results using the proposed algorithm on a set of synthetic images and a sequence of real images are presented.

4.1.2 Line matching

Our approach for the computation of a 3D line set, using oriented images and extracted 2D lines, closely follows the one described by Schmid and Zisserman [113]. Multiple images of a scene together with extracted 2D line segments are used to compute and verify 3D line hypotheses. The algorithm uses both, the geometrical information of the 2D line segments and the photometric information of the corresponding images. The result of the line matching process is a set of 3D line segments in object space.

4.1.3 Plane sweeping

The process of plane sweeping is discussed in [140] and [5] and will only be briefly sketched here. The main difference is the use of a similarity function based on edgel point sets instead of computing the similarity via image correlation for interest points. The 3D lines are used to generate planes hypothesis from which the plane homographies between image pairs are computed. The homographies induced by the sweeping plane are represented as a 3×3 homography matrix H and allow direct point transform between images that view a planar region. This means that the edgel point sets A of the planar region captured by the first cameras image can now be directly transformed to its corresponding view $\tilde{A}$ for the second cameras image. This is done by multiplication with the given homography matrix: $\tilde{A} = HA$.

The homographies are computed for a 3D plane that is observed from two different views. A detailed description an be found in [50] pp. 325 ff. The plane is defined by its normal vector n and distance to origin d.

4.1.3.1 A similarity measure for point sets

With the homographies given, we are able to transfer edgel sets in the vicinity of the projected 3D lines and measure the degree of similarity. The similarity between two point sets $\tilde{A}$ and B, where $\tilde{A}$ is the transformed edgel point set of the first image, is measured by counting the number of points in $\tilde{A}$ that have a corresponding point in B within a predefined radius r. The same measure is computed for $\tilde{B}$ - in this case B is mapped with the inverse homography. This approach makes the measure robust against segmentation errors and clutter. A further increas in robustness can be achieved when the orientation of the 2D features is also taken into account. For typical sweeping scenarios the radius r is in the range $0.5 \ldots 5$ pixel. For fast location of the nearest neighbor the 2D edgel sets are stored in a KD-tree.

4.1.3.2 Sweeping modes

In this subsection the different sweeping modes are explained. There are two possible sweeping modes: a rotational sweep mode and a translational sweep mode. Figure 4.2 shows an illustration of the two modes. The rotational sweep in this example is performed for the eave line of the captured facade (the camera is symbolized as a pyramid) and the translational sweep is performed for the whole facade.

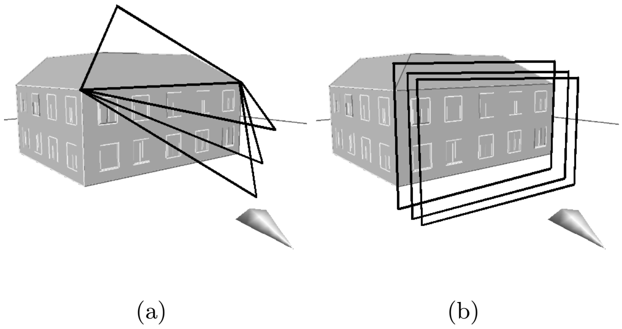


Figure 4.2: Illustration of the two sweeping modes (the camera is symbolized by its frustum). (a) rotational sweep planes for the eave line of the facade; (b) translational sweep planes;

Rotational sweep In case of the rotational sweep a 3D line serves as rotational axis for a set of plane hypothesis. The 2D edgels which are transformed during the sweeping process are taken from the neighborhood of the projected 3D line from the first image and actually split into two edgel sets on both sides of this 2D line are extracted. Assuming a planar patch on at least one side of the line, one of the two edgel sets will be aligned.

In the next step an initial hypothetical sweeping plane using the 3D line and the origin of the camera is instantiated. In the following sweeping step the plane is rotated around the axis formed by the two points of the 3D line. In order to avoid numerically instable conditions at the start of the sweeping process, when the plane is viewed edge on, it is first rotated to a reasonable start position by a fixed angle φ_{min}. This condition is also met at the end, so the sweep angle range for the plane is $\varphi_{min} \leq \varphi \leq \pi - \varphi_{min}$. The plane hypothesis at the start position, two of the subsequent planes and the end position are shown in Figure 4.2(a). Generally the rotatonal sweep is of lesser importance due to the fact that it needs 3D liness for initialization and the achievable accuracy strongly depends on the accuracy of the particular line that defines the sweeping plane.

Translational sweep The translational sweep is typically used to detect dominant facade planes. For the translational sweep the plane hypothesis is not induced by a 3D line but by a vector that defines the sweeping direction. This sweeping direction can be computed from 3D lines by clustering, but can also be determined from two known vanishing points [108]. A pair of orthogonal vanishing points is a strong indicator for the existence of a 3D plane, but non orthogonal vanishing points may also belong to a planar structure (think e.g. of floor tilings, diagonal elements in a facade etc.).

The input for the clustering are crossproduct vectors from 3D line pairs, where only line pairs with a reasonable large inner product are considered - this ensures that lines that point into the same general direction are not used to generate vectors for the clustering step.

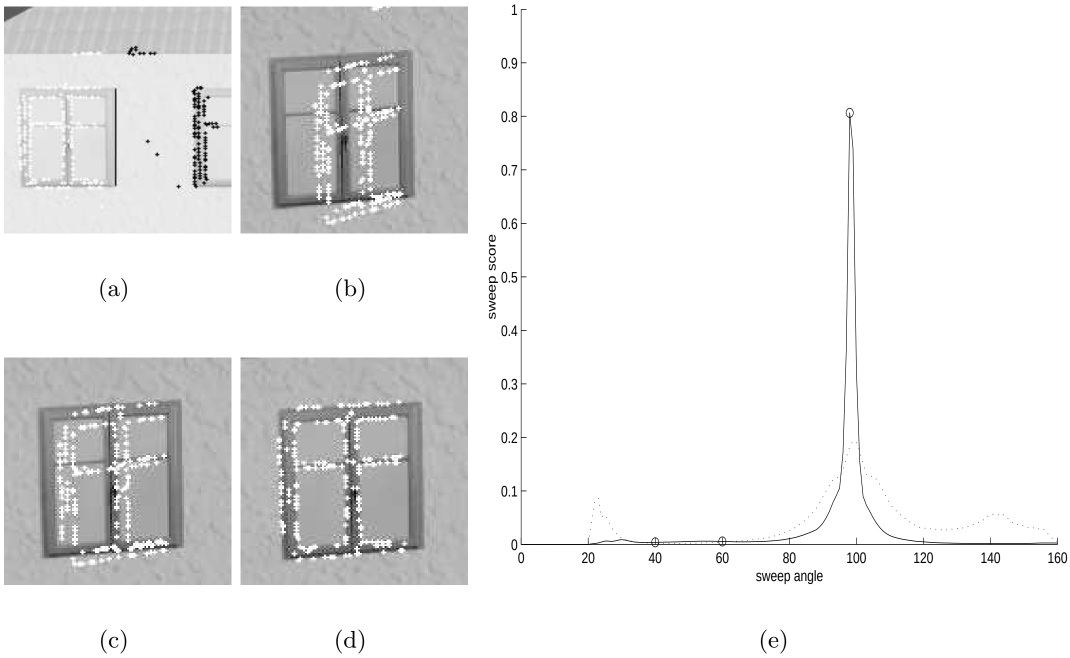
In contradiction to the rotational sweep, where the sweep range is bounded, the translational sweep is only bounded at one side, the camera origin. The second bound must be determined from the scale of the observed scene e.g. from reconstructed 3D points from the image orientation stage. This initial plane is then swept along the direction of the normal vector. Figure 4.2(b) shows three plane hypothesis for the dominant facade.

4.1.3.3 Sources of error

The orientation of the detected planes in sweeping mode is sometimes slightly inaccurate, caused by the fact that detected 3D lines do not lie exactly on the plane. In case of architectural models 2D lines often are detected on topological features that stand out of the facade plane such as window sills and friezes or lie behind the facade plane e.g. window frames. In this case the final orientation of the plane does not match the normal vector of the facade. A typical case is a 3D line that is detected on a protruding window sill; The best alignment that can be found for the plane is somewhere on the facade and therefore incorrect. Another problem are 3D lines with an inaccurate orientation, for those lines the planes are also slightly inaccurate. With the translative sweeping approach it is possible to overcome the problems described above, since the normal vector for a translative sweep is computed by clustering.

4.1.4 Experiments

We conducted several experiments with synthetic as well as real image data. In Figure 4.3 the sweeping process for synthetic data is shown. Figure 4.4 shows the detected plane for the synthetic data.



(a) (b)

(c) (d) (e)

Figure 4.3: Alignment of edgel sets for a 3D line. (a) template edgel sets left (light crosses) and right (dark crosses)from a projected 3D line. (b) transformed left edgel set at 40^o and (c) 60^o (d) final alignment of the left edgel point set (d) sweep scores for both edgel point sets (dashed and solid) over the sweep range ($20^o - 160^o$).

The next experiment compares the image-based methods versus the feature-based approach. Figure 4.5 shows the score values plotted against the sweep angle. The score for the feature-based method is plotted solid, the score for the image based method is dotted. Multiple maxima, caused by repeating patterns on the facade, make it difficult to detect the best maximum for the image based approach. The score function for the image based approach gets smoother if a larger correlation window is used, in our experiments the size of the correlation window was 19.

Accuracy investigations complete the series of experiments. These experiment was carried out using synthetic data which known ground truth. Table 4.1.4 shows a comparison of the image-based method versus the feature-based approach. The error between the normal vector of the detected plane and the correct normal vector is used to compare the accuracy of the two methods. The experiments were carried out on images corrupted with different levels of Gaussian noise using 23 planes. The feature-based method performs better than the image-based algorithm especially if the images are corrupted with noise. Our method is about twelve times faster than the image-based approach, though there

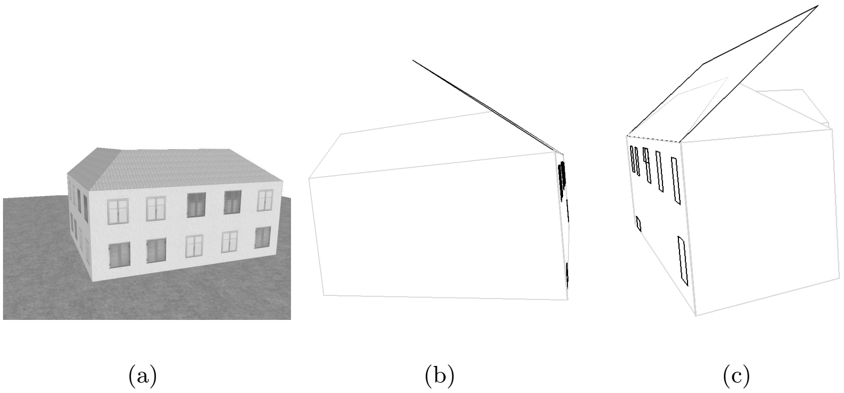


Figure 4.4: Synthetic input data and resulting plane hypotheses, note the accurate alignment of the planes. (a) synthetic input image; (b) view from the left side; (c) view from the right side.

	% noise	mean error	std. deviation
Feature-based	0	0.8736	1.1087
Image-based	0	2.0568	7.52
Feature-based	2	0.9216	0.9873
Image-based	2	2.0378	8.1545
Feature-based	5	1.4808	5.4877
Image-based	5	3.2758	11.4760

Table 4.1: Feature-based method versus image-based method. The error is the difference between correct normal vector and the normal vector of the detected plane. (23 planes were used in this experiment)

was no effort made to optimize the image based method for speed.

We presented an improved method for plane hypothesis generation using a feature-based plane sweeping approach. The experiments showed that the feature-based approach is faster than the image-based method and yields also more accurate results. As mentioned in subsection 4.1.3.3 errors are mainly caused by 3D lines that are not lying on the plane. A solution for this problem could be a final alignment of the point set with a nonlinear optimization method that performs an estimation of the 3D parameters based on the detected point correspondences. Such aa approach is proposed by Fitzgibbon [36].

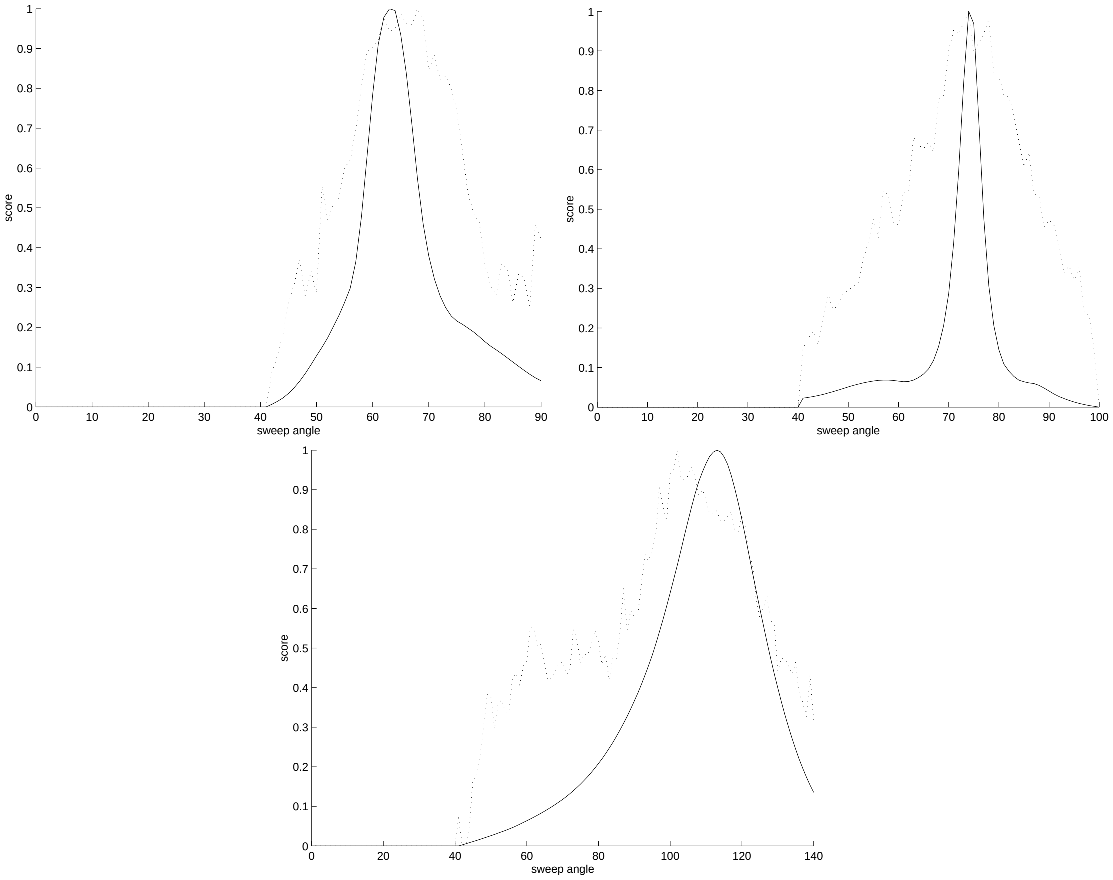


Figure 4.5: Comparison of sweep scores of new edgel-based approach and the image-based approach. The scores for the image-based approach are dashed and the scores of the new edgel-based method are shown as continuous line. Note the significantly sharper peak for the feature-based method.

4.2 Sparse 3D reconstruction by edgel-based space-sweeping

In this section a method for the efficient and robust generation of directed 3D primitives from 2D edgel observations is presented. The description of this method concludes the theoretical part of this theses and should be considered the main contribution of the work. From the first implementation to its present state the proposed method underwent several iterations which all resulted in an improvement in robustness as well as computational efficiency. In short a feature-based method for the fast generation of sparse 3D point clouds from multiple images with known exterior orientation is presented. The proposed approach works solely on directed 2D primitives (e.g. edgels or ridgels). These geometric image features are described by their 2D position plus an associated orientation vector. Edgels and ridgels can be extracted with sub-pixel accuracy from the given input images. A so called space-sweeping scheme is used to compute the accurate 3D location of these edge features. The proposed approach relies mainly on the geometric properties of the extracted primitives and incorporates a robust uncertainty estimation scheme to detect outliers.

4.2.1 Introduction

The computation of 3D structure from multiple images is one of the most important tasks in computer vision. In literature many different approaches have been described. Many of the basic methods are working with image tuples, the stereo pair. Especially many dense stereo matching methods were formulated for stereo image pairs. Recent methods [65], incorporate multiple images to achieve a more robust matching result. Correlation-based dense stereo matching methods are usually restricted to small baseline setups and in the case of video-based stereo to image sequences with very large overlap factors. These dense image matching methods compute a disparity map for the scene, i.e. for every pixel a disparity vector $d(x, y)$ is given. The disparity is often constrained by a smoothness criterion and the ordering constraint. The negative effect of the smoothness constraint is that sharp creases or depth-discontinuities appear smooth in the reconstructed model. Another branch are the various voxel coloring methods [72, 115]. Those methods produce a volumetric model of the scene and also work for scenes where the ordering constraint is violated. All the above mentioned reconstruction approaches have the drawback of processing times in the order of minutes. Even though if the processing time is reduced by hardware implementations of the algorithms (nowadays especially on graphics hardware) the complexity may still be prohibitive.

The method that is going to be presented in the following produces sparse clouds of 3D primitives. Each primitive is characterized by a 3D position and an associated orientation vector. Such clouds of 3D primitives are useful for model-based reconstruction. For example the extraction of 3D lines, the detection of 3D planes or other 3D primitives can be based on sparse 3D data. Furthermore the proposed approach yields accurate measurements on depth discontinuities and thus is suitable to enrich the fidelity of reconstructions carried out by standard dense matching/modeling approaches. One can think of using the sparse 3D primitives as robust seed points for a subsequent dense reconstruction.

The introduction of the space sweeping method is structured as follows: At first an introduction to the general plane sweeping strategy is given and the key publications are discussed. The next subsection gives a more detailed presentation of the implementation of the method and an in-depth analysis of the spatial data structures involved in the hypothesis collection stage. The topic of the third subsection is the 'weeding' stage, namely the selection of valid hypothesis. We discuss the use of an image-based similarity measure to verify the resulting 3D hypotheses and propose an error propagation scheme to verify the hypotheses using purely geometric criteria. The fourth and final subsection proposes the use of contour chains as higher level primitives combined with an energy minimization scheme in order to improve the robustness of the method. This subsection also provides concluding remarks and an outlook on future work concerning this part of the theses.

4.2.2 Space-sweeping methods

Space sweeping methods are a common way to realize a multi-image 3D reconstruction approach. Collins [26] introduced a feature-based space sweeping approach for sparse 3D reconstruction and formed the notion of true multi-image matching methods. These methods should:

1. Generalize to any number of images greater than two;
2. Have an algorithmic complexity that is linear in the number of images;
3. All images should be treated equally;

While condition one is met by many reconstruction approaches, condition two addresses the efficiency of the reconstruction method: if it repeatedly processes a subset of images and then fuses the results it can, according to Collins, not be considered a true multi-image reconstruction scheme. However, a valid strategy is the sequential processing of image pairs and a subsequent fusion stage. The third condition, namely the equal treatment of all images can only be met by algorithms working in object space.

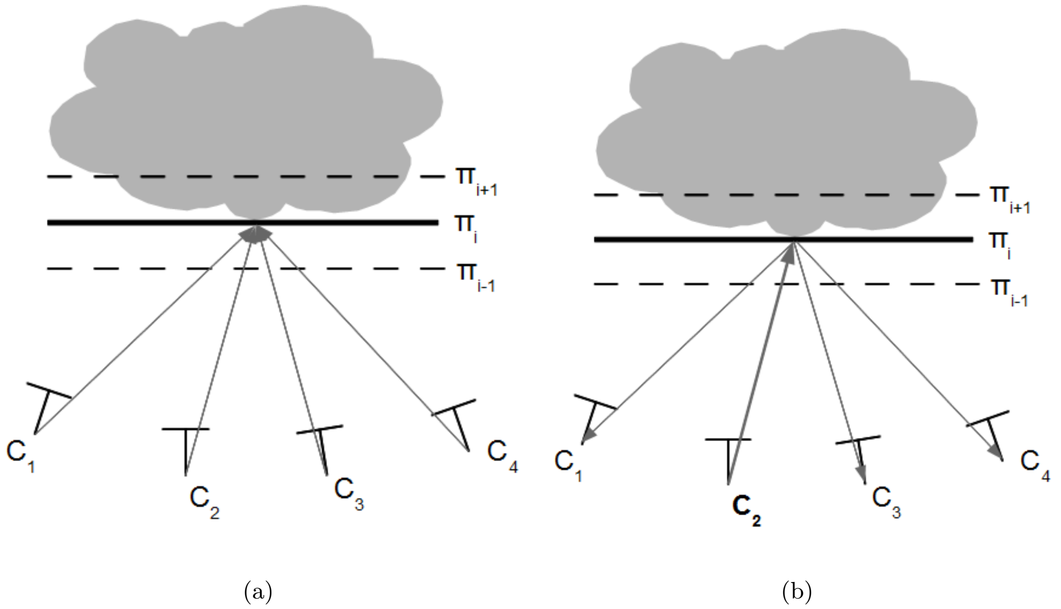


Figure 4.6: Top view of the space-sweep setup: Four cameras (symbolized by their image plane and camera origins $O_1 \dots O_4$) view an object (light gray). The sweep plane moves from front to back, the previous and subsequent instance of the sweep plane is illustrated as dashed line. The on left the space sweeping method as proposed by Collins [26] is shown. In this case all cameras are treated equally and the detection and verification of 3D hypotheses happens in the object space. The right sketch shows the proposed approach where the detection and verification of hypotheses is preformed in the image space. A reference camera (origin O_2) is chosen and from this camera a 3D ray (shown as bold vector) is intersected with the sweeping plane (bold line). The verification of 3D hypotheses is achieved by projecting the resulting 3D point into all slave-cameras.

In general a space sweeping method is used to traverse the volume for which the 3D reconstruction should be performed. Figure 4.6(a) and (b) show the top views of such a setup. Figure 4.6(a) shows the space sweeping approach of Collins - here the hypothesis verification happens in object space - meaning that all cameras are treated equally. Multiple cameras $C_1 \dots C_4$ (symbolized by their image plane) view an object (light gray). A virtual sweeping plane π traverses the volume from front to back and 3D hypotheses are gathered by intersecting rays emanating from the cameras with the sweeping plane and evaluating a proximity-based criterion. Collins uses a 3D accumulator plane (a plane partitioned into cells) and increases the accumulator count of the cells that are closest to the intersection location. This way every cell counts the number of rays that coincide in a discrete volume in space. If a sufficient number of votes from different cameras is encountered in an accumulator cell a tentative 3D hypothesis is recorded. Subsequently a simple statistical model of clutter is used to determine whether a hypothesis, formed by the intersection of several rays, is a valid one. Despite the elegant formulation of the multi-image matching problem the main drawbacks of Collin's approach are the expensive hypothesis evaluation based on the 3D accumulator plane and the fact that geometrical properties of the 2D image features, such as edge orientation or principal directions of corners, are neglected.

Another class of space sweeping algorithms are the voxel coloring approaches that also work in object space: These methods produce a volumetric reconstruction of the scene. The reconstruction volume is represented as discrete voxel space and a image-based similarity criterion, namely the color consistency, is used to discard all voxels that lie in front of a surface point. All voxels that are invisible in all views are assumed to belong to the actual object volume and are therefore retained.

In figure 4.6(b) one camera (C_2) is chosen as key/reference camera. For all features that are extracted in the image of this camera the corresponding rays are intersected with the 3D sweeping plane. The resulting 3D point is projected into all other slave-cameras for hypothesis gathering and verification. It is obvious that criterion three of Collin's true multi-image reconstruction postulate is violated, however the hypothesis verification in image space has advantages over the object space approach: Efficient 2D spatial data structures can be used to evaluate the proximity criterion and various geometric criteria and image-based similarity criteria (e.g. normalized cross correlation) can be used for the robust selection of 3D hypothesis. An example is the method proposed by Jung et al. [64] which computes sparse 3D point clouds from multiple oriented images. The method is used for extracting 3D information from aerial images. It uses edgels that are extracted with sub-pixel accuracy from the input images and are then transformed into 3D rays for the intersection with the sweeping plane. The intersections are projected into the slave images and an efficient quad tree search is used to determine a set of 3D hypotheses. All hypotheses are then refined using a least-median-of-squares technique. Finally the 3D direction of the point hypothesis is computed using the 2D direction associated to the edgels that support an individual 3D hypothesis in the following way: Planes are formed by the edgels position, the edgels direction and the camera center. The pairwise intersection of the planes from the different cameras gives a bundle of possible directions. From this bundle the best direction is determined by a LMS technique.

The approach that is proposed in this thesis is inspired by [64], but differs from this work in several aspects:

1. Efficient spatial data structures are employed to speed up the search for tentative 3D hypotheses.

2. The 3D reconstruction is based on directed 2D primitives instead of point primitives.

3. Contour chains are used to provide pre-segmented input data.

4. An energy minimization scheme is used to determine outliers within a contour chain.

5. While the original method [64] is used for reconstructing buildings from aerial images the proposed method is applied for close range scenes.

The following a detailed description of the individual steps of the proposed approach is given. The principal work-flow of our approach can be outlined by five consecutive steps:

1. Feature extraction: Directed 2D primitives are detected with sub-pixel accuracy. These can be edgels or ridgels.

2. Space Sweeping: 3D rays are formed for all 2D features in the reference camera and used to select candidate features in the slave images i.e. features that lie close to the reprojection of the 3D search ray.

3. Generation of 3D hypothesis: 3D planes are computed for the candidate features that are found in the slave images and used to compute tentative 3D hypothesis.

4. An error propagation scheme is used to estimate the reconstruction uncertainty in object (3D) space. The detection of outliers is based on the evaluation of the reconstruction uncertainty.

5. Optionally an energy minimization scheme can be applied to eliminate remaining outliers. This step assumes that individual 2D features are linked to chains and applies a smoothness criterion on the reconstructed object points.

In the following subsections we will explain the methods in more detail.

4.2.3 Feature extraction

The proposed approach uses 2D point primitives with an associated 2D direction. This can be either edgels or ridgels which are extracted with sub-pixel accuracy. Furthermore a robust gradient direction must be associated with each edgel. In order to capture also very fine detail ridgels can be used, however the direction associated with ridgels is ambiguous (see 2.3) and these primitives have to be treated differently in the sweeping approach. Edgels and ridgels capture most of the prominent features of typical urban scenes such as structural and textural elements of facades. The extracted primitives can also be grouped into contour chains and this higher level knowledge can then be used for detecting outliers in a post-processing step. such a grouping step, where only chains with a certain number of supporting primitives are accepted, also helps to remove isolated primitives that are often generated by small structures and therefore have an unreliable direction assigned with them.

Harris corners are another class of 2D primitives that can be extracted with sub-pixel accuracy, however the estimated principal direction (e.g. one of the two eigenvectors of the structure tensor) is corrupted by significantly more noise than it is for edgels or ridgels. Due to the sparseness of the extracted corners a robust grouping is not possible and as a consequence corners are not used in this approach.

4.2.4 Space sweep

As mentioned in the introduction a good hypothesis is characterized by a low reprojection error, i.e. the 2D image location of the reprojected 3D point is close to a feature point in all or many of the slave-cameras. The evaluation of the proximity criterion is the most costly step in the sweeping framework, since it involves reprojection of the 3D hypothesis into all slave images and the determination of the closest neighboring edgel to the reprojected hypothesis. Essentially this boils down to finding candidate features in all slave images by performing a nearest neighbor search for the reprojected 3D point. In the following an overview of spatial data structures for an efficient nearest neighbor search is given.

4.2.4.1 Candidate selection

In the sweeping stage the initial selection of hypotheses is primarily based on the proximity criterion. In the original paper of Jung et al. [64] the proximity criterion is evaluated using a quad-tree for the nearest neighbor search. The quad-tree approach, however, has a computational complexity of $O(log\ n)$, where n is the number of points in the set, per nearest neighbor query. A data structure of the same computational complexity is the KD-tree.

A more efficient method is the class of distance transforms. The distance transform is computed for a set of 2D points and can be considered as an image where the value of each pixel encodes the distance to its closest point. Distance transforms can be computed efficiently in linear time using Chamfer filtering [20]. In order to keep the relationship between the points and the distance image an additional label image is created. Figure 4.7 shows the 2D source points on the left, the distance transform image in the middle and the label image on the right.

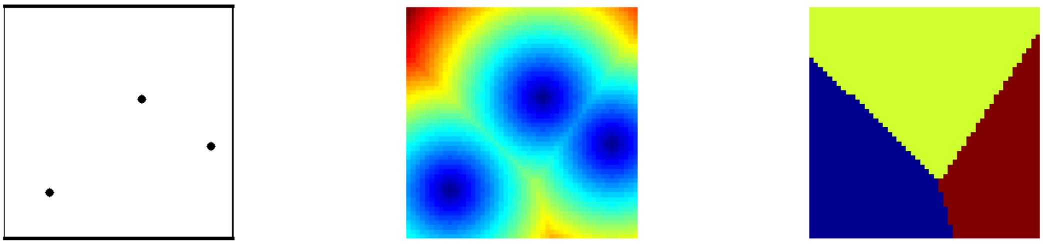


Figure 4.7: Illustration of the distance transform: For a set of 2D input points (left) the distance transform encodes the distance to the closest points and the label image (right) holds the index to the corresponding point or in other words the pixel value refers to the index of the closest point. The distance transform image and the label image allow for fast nearest neighbor queries.

The distance transform allows nearest neighbor queries for 2D point sets in $O(1)$. For the plane sweeping a separate distance transform image is computed for the edgel locations of every slave image. Since the 2D coordinates of the reprojection are non-integers the image access operation in the distance transform images is performed using either with bi-linear or with bi-cubic interpolation. An experiment where the distance transform approach was compared with an exact KD-tree-based nearest neighbor search showed that an average distance error of less than 0.6 pixel can be achieved. This shows that it is justified to use the much faster distance transforms to generate an initial set of 3D hypotheses. However the memory consumption of distance transforms is not negligible and despite the fact that the nearest neighbor queries for 2D point sets can be performed in $O(1)$, the computational overhead is considerable.

A disadvantage of all aforementioned methods is that the detection of hypothesis involves a linear search along the reprojected 3D search ray in all the slave images. Figure 4.8 illustrates the search range of a reprojected 3D ray for two slave cameras. The 3D ray is defined by the 2D position of a feature in the reference image and the reference cameras exterior orientation. A predefined volume of interest or another estimate for the depth range of the observed scene are used to define a minimal distance d_{min} and maximal distance d_{max} that delimit the ray. The search space for candidate features in the slave images is now a 2D line segment (shown as dashed line) that is generated by the projection of the bounded 3D ray.

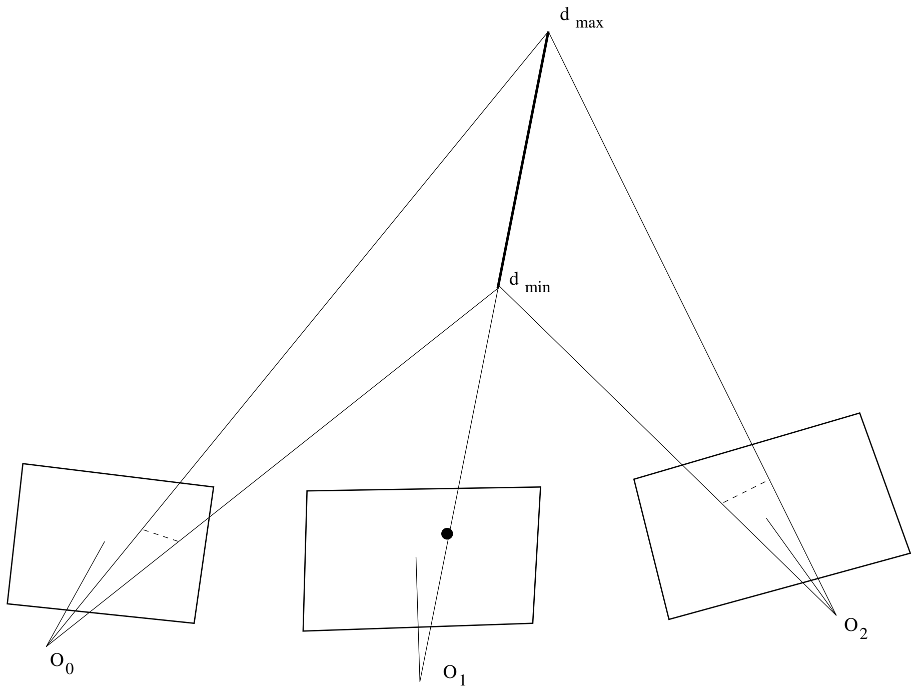


Figure 4.8: Illustration of the search range that is generated by reprojection of a bounded 3D ray that emerges from the reference camera (with origin O_1 and illustrated as thick line ranging from d_{min} to d_{max}) into two slave images (with origins O_0 and O_2). The reprojected ray (the epipolar line) is shown as dashed line in the slave images.

A direct approach that uses distance transforms would now perform a linear search along the 2D lines in order to find candidate features. For close range scenarios, where the relative rotation between camera pairs can be large, the projection of the 3D search ray can become quite long. For a high resolution image the reprojection of a typical search ray can span several hundred pixels. The necessity for performing a linear search implies that the complexity of the hypotheses generation process depends not only on the data (the 2D primitives) only, but also on the geometrical configuration of the cameras.

As a consequence a more efficient search strategy based on stereo rectification was devised. In order to accelerate the spatial search for edgels lying close to the reprojection of the 3D search ray a pairwise stereo rectification between the reference camera and the particular slave cameras is performed. The stereo rectification determines a perspective plane-to-plane transform (homography) for each image such that all pairs of matching epipolar lines become collinear and parallel to the x-axis. In the case of n input images $I_1 \dots I_n$ one ends up with $n-1$ pairs of homography transforms $(H_{k_i}, H_i), i = 1 \dots n | i \neq k$ where H_{k_i} is the transform matrix to achieve the stereo rectification for the reference image I_k with the slave image I_i and H_i is the homography matrix for the slave image. All extracted 2D features, except those of the reference image I_k, are transformed with the respective homography transform. Note that also the principal direction associated to the features has to undergo this transform. Additionally the transformed features are sorted with respect to their y-coordinate. This sorting allows a very efficient query for features that lie close to an epipolar line.

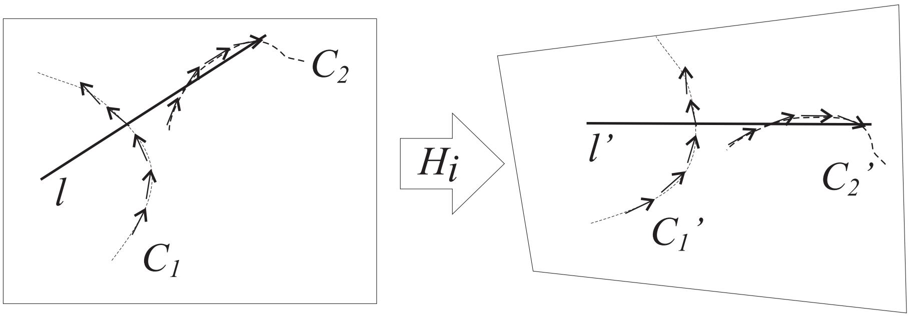


Figure 4.9: Illustration of stereo rectification: The 2D features in the original image I_i (left) are transformed by the homography H_i. After the transformation the epipolar line l' is parallel to the y-axis (and so is every reprojection of the 3D search rays). In the rectified image it is now possible to filter out features with a tangent vector that is nearly parallel to the epipolar line (l in the original image and l' in the rectified frame). The edgels of contour C_2 would therefore not be used in the sweeping process.

An additional advantageous side-effect of the rectification is that features with a tangent vector that is nearly parallel to the epipolar line can easily be identified and excluded from sweeping process. Figure 4.9 illustrates this elimination process for the features of two contours C_1 and C_2: The tangent directions of the feature points of C_1 have a significantly larger enclosed angle with the epipolar line l than the features of contour C_2. In this case all features of C_2 would be discarded in the sweeping process. Due to the rectification process the detection of features whose tangent direction is nearly parallel to the epipolar line can be achieved by simply testing the magnitude of the y-coordinate of the normalized tangent direction (or the x-direction of the gradient vector) of the geometric primitives. The filtering is carried out with a threshold of 10 degrees in all our test cases. This simple strategy helps to avoid ambiguous situations, where many primitives with a tangent direction that is nearly parallel to an epipolar line segment prevent the detection of a distinct 3D hypothesis.

The rectification process introduces a perspective distortion that transforms the rectangular image plane into a general convex polygon. In order to overcome the effect of additional uncertainty in location accuracy introduced by the rectifying transform, the relation between original features and rectified primitives is kept and for all 3D reconstruction steps the rays/planes formed by the original primitives are used. We use the rectification method proposed by Fusiello et al. [42]. In contrast to image-based matching approaches the rectification method needs not to be optimal with regard to the introduced distortion, since we keep the relationship between original and rectified features.

The space sweeping is performed by computing a 3D search ray r for each feature in the reference image I_k. The bounding points of the 3D ray are defined by the volume of interest (or a front and back plane that bounds the depth range).

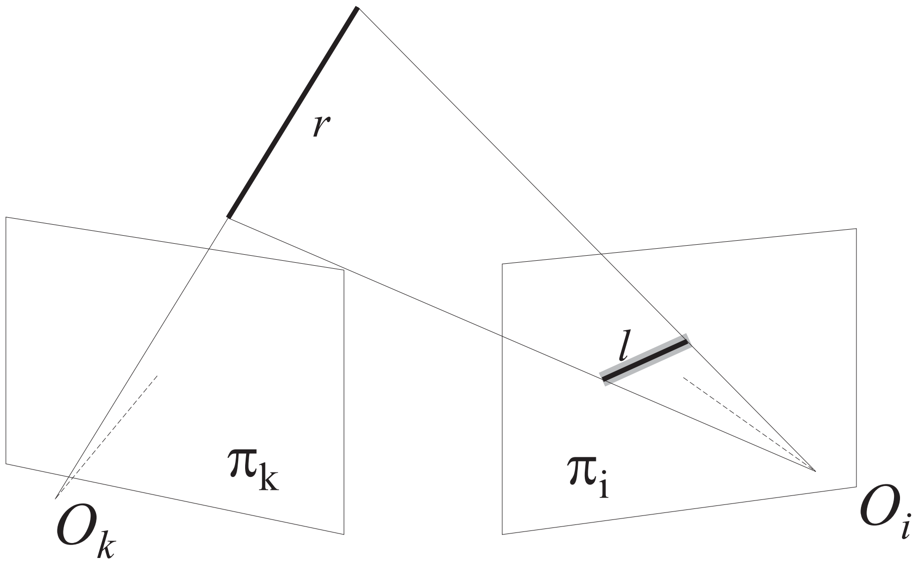


Figure 4.10: Illustration of the space sweeping principle for a reference camera (image plane π_k with origin O_k) and a slave camera (image plane π_i with origin O_i): A 3D line segment r (a portion of the 3D search ray) is projected into the slave image and forms an epipolar line segment. All edgels that lie within the gray shaded area are considered potential candidates.

Figure 4.10 shows the principle: the 3D line segment r is projected into all slave images in order to detect features that lie within distance d (gray shaded area) of the epipolar line segment. Since the features are sorted with respect to their y-coordinate and the epipolar line is parallel to the x-axis, the range query for candidate features can now be accomplished in a very efficient manner. Basically a binary search detects the upper and lower bound for valid feature points. However, the search rectangle in the un-rectified image plane is distorted to a convex four point polygon in the rectified image plane. We use the bounding rectangle of this polygon do determine a conservative axis-aligned box in which we search for candidate edgels. A typical threshold-distance d used in all our experiments is 0.5 pixel. Figure 4.11 illustrates this spatial query with an epipolar line segment.

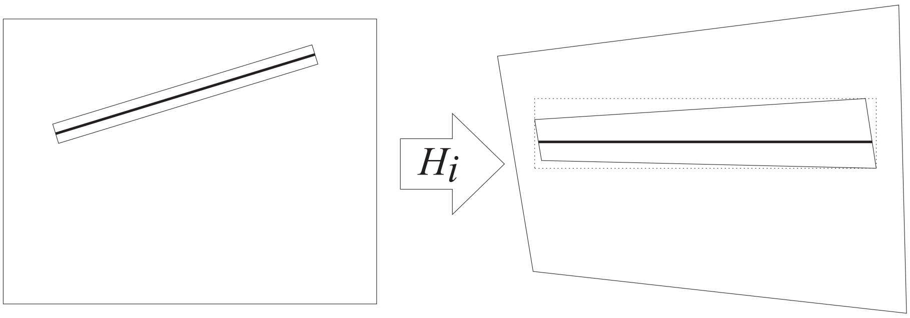


Figure 4.11: Nearest neighbor queries for rectified image planes: The rectangle defined by the epipolar line (continuous line) in the original image is shown on the left. The homography H_i transforms the rectangle into a convex four point polygon in the rectified image plane on the right. A conservative axis-aligned bounding box (shown with a dashed outline) is used to collect the feature candidates.

Here another advantage of the stereo rectification comes into play: we can enforce that the feature in the reference image (that forms the 3D search ray) and the candidate features have the same principal direction. This means that if the edgel in the reference image is formed by a bright-to-dark edge all candidate edgels in the slave images must also be formed by a bright-to-dark edge. This constraint resolves many of the ambiguous situations that occur when only the proximity of the edgels to the epipolar line is considered. As a result of the space sweeping a set of candidate features is found in each of the slave images. This strategy can not be used for ridgels due to their orientation ambiguity. All candidate features that are found by this efficient search are then used to generate 3D hypotheses in the subsequent 3D hypothesis generation process.

4.2.4.2 3D hypothesis generation

In the 3D hypothesis generation step we treat the 2D primitives no longer as oriented point-like features, but use the associated tangent directions to form 3D planes for all candidates. These 3D planes are then intersected with the 3D search ray from the reference camera and thereby generate depth events for all candidates. This strategy allows to transform all candidates into the common reference frame of the 3D object coordinate system. The generation of depth events is performed for all candidate edgels from all slave images.

Figure 4.12 illustrates how the tangent direction of a 2D primitive is used to form a depth event: The feature position (black dot) and the tangent direction of a valid candidate feature form a 3D plane π_e, which is intersected by the 3D search ray r_k at v (marked by a circle). The introduction of depth events allows now to impose an ordering on the 3D planes π with respect to the 3D query ray r_k.

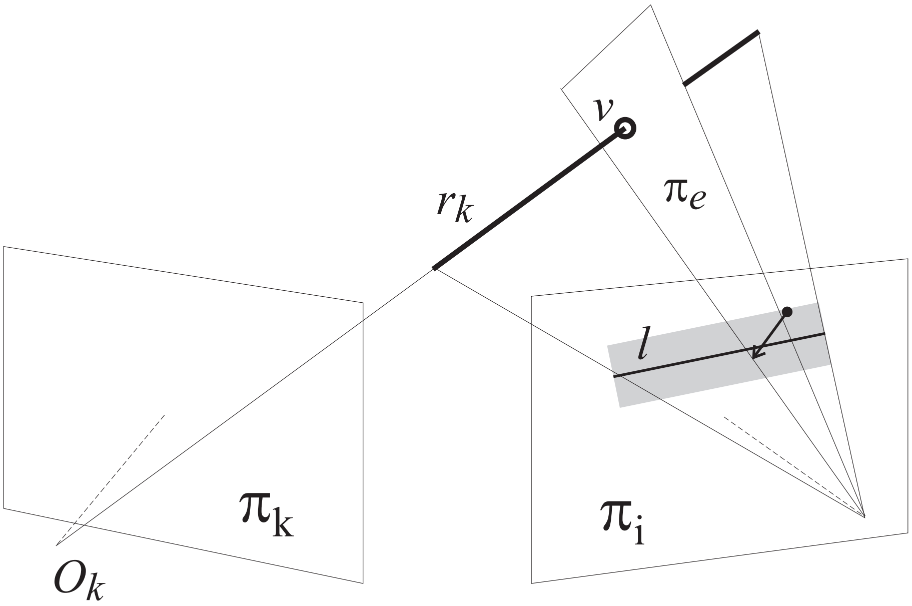


Figure 4.12: Generation of depth events that lie exactly on the 3D search ray r_k. The feature location (black dot) and the tangent vector (vector intersecting the epipolar line l) form the plane π_e and the intersection of π_e and r_k generates the depth event v (shown as circle).

Thus, for a particular 3D search ray r_k a number of depth events is generated and the events are sorted with respect to their distance from the reference cameras origin O_k. This sorting of depth events according to their depth from the reference cameras origin O_k allows the efficient detection of clusters, where rays from different cameras intersect, to be achieved in an event-driven manner. The actual 3D hypotheses are found by searching for clusters of depth events on the ray r_k. In order to generate a valid 3D hypothesis, depth events from $m < n$ (where n is the total number of cameras) different cameras must vote for a similar depth. For the event-driven clustering a 1D range query for all depth events along the search ray is performed. If sufficiently many depth events from different cameras can be found within a given depth range, the current depth event is considered a tentative 3D hypothesis. The depth range is a user specified parameter that determines how closely packed the depth events must be in order to form a valid 3D hypothesis. A possible approximation of the depth range must consider the relative pose of the cameras, the size of the reconstructed volume as well as the overall accuracy of the camera pose estimation and feature extraction. Due to the sorting of the depth events the complexity of clustering is reduced to a 1D range query. Figure 4.13 shows an illustration for the detection of 3d hypotheses on a 3D search ray.

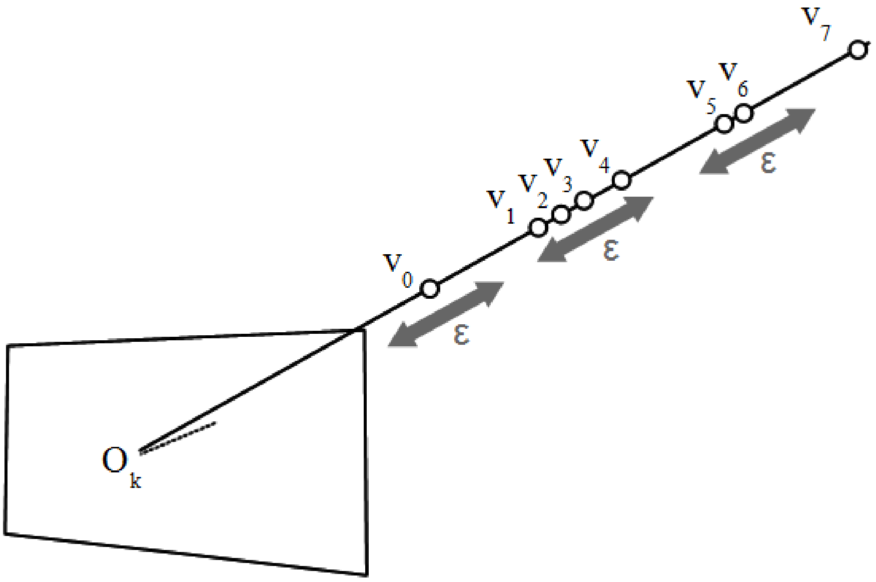


Figure 4.13: Illustration of the detection of 3D hypotheses on a 3D search ray. A number of depth events $v_0 \dots v_7$ lie on the 3D ray. The depth events are sorted according to their distance from the reference camera's origin O_k. The criterion for a valid 3D hypothesis is that $m < n$ depth events from different slave cameras form a cluster within the predefined range ε. Thus, for $m = 2$, two 3D hypothesis are found, the first for the depth events $v_1 \dots v_4$ and the second for the depth events v_5, v_6. The depth events v_0 and v_7 are too far from any neighboring event to form a valid hypothesis.

For all detected clusters with sufficiently many votes from depth events, we compute the 3D line segment that minimizes the sum of squared perpendicular distances to the planes

π_{e_i} that are associated with the depth events v_i. This is an efficient linear operation and is described in more detail in [50] (p323). This approach avoids the separate estimation of the 3D position (based on multiple ray intersection) and the 3D direction as it is proposed in [64]. The resulting 3D point is defined by the intersection of the reconstructed 3D line and the 3D search ray.

Briefly summarized the search for a valid 3D hypothesis along a particular 3D ray is now reduced to the detection of clusters of depth events. The events must be spatially close and be related to different slave cameras. The 3D primitive is reconstructed from all the 3D planes that are associated to the depth events by a linear method. Despite the fact that only one physically correct depth value does exist for a particular 3D ray, we reconstruct and store all 3D hypothesis and perform a subsequent hypotheses selection step.

4.2.5 3D hypothesis verification

The sweeping process is designed to detect all tentative 3D hypotheses on the 3D sweep rays. This strategy implies that it is possible to find multiple hypothesis along a particular search ray. The number of ambiguous hits depends on the observed scene as well as the geometric configuration of the camera poses. Cluttered scenes where many features are detected or objects that exhibit recurring patterns lead to ambiguous situations, where several equally ranked hypotheses are found along a particular search ray. Performing the sweeping for typical close range scenes with a high depth range, might also increase the number of hypotheses on a search ray.

The task of the 3D hypothesis verification is to find the one hypothesis which most likely describes the correct physical point.

For the detection of these hypotheses two methods have been devised. The first method uses the image information around the candidate features to judge whether a 3D hypotheses is accepted or not.

4.2.5.1 Image-based Outlier Removal

In order to detect and remove false positive matches, an image-based similarity measure is applied: The normalized cross correlation (NCC) is computed for image patches that are sampled around the candidate features. In order to achieve a robust similarity measure for a 3D hypothesis the local patches are sampled in a rotation and scale invariant way. Scale invariance is achieved due to the fact that the 3D coordinate of the hypothesis is known.

The scale of the patch can be directly computed from the 3D hypothesis' distance from the camera. Rotation invariance is achieved due to the known directional component of the hypothesis - remember that the hypothesis is constructed from is used to define a reference direction in the image space. Bearing in mind that the 3D hypotheses are often formed by edgels lying on depth discontinuities the reference direction is used to split the image region into two separate patches. These makes the similarity criterion also insensitive against occlusions. Figure 4.14 shows the generation of rotation and scale invariant descriptors for candidate edgels in two images: The image position of the reprojected primitive (shown as disc) and its associated tangent direction (continuous line) is used to divide the local region and to sample the two patches on opposite sides (shown with a different hatching). Due to a depth discontinuity the lower right patch in the two image regions is occluded. In such a case only one side can be used for reliable similarity comparison.

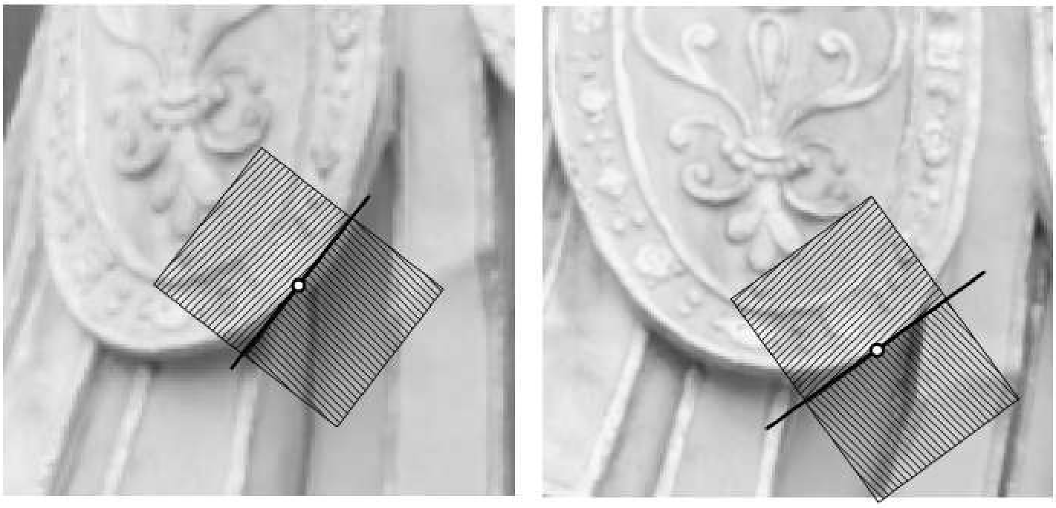


Figure 4.14: Sampling of a scale and rotation invariant patches. Reprojecting a 3D hypothesis into the images allows to sample a rotation and scale invariant patch. The 2D image position (shown as disc) and the 2D direction of the projected hypothesis define the location and orientation of the patch. The distance of the hypothesis from the respective camera origin defines the scale. The 2D line that is defined by the position and orientation of the primitive (shown as continuous line) is used to split the patch in half and sample two descriptors (illustrated with a different hatching). This approach increases the robustness for hypotheses that lie on depth discontinuities.

Therefore we compute the normalized cross correlation separately for the two image regions between the reference image and the slave images and take only the higher correlation value into account. The correlation based similarity measure is performed in all images with a contributing image feature. The final score is then computed as the average correlation value over all contributing images. The 3D hypothesis with an average correlation value above a given threshold (0.8 in our experiments) is finally accepted. A drawback of this image-based similarity criterion is the fact that the detected primitives are often located on borders between homogeneous regions. Therefore the window size of the sampled patches has to be sufficiently large in order to allow distinction. Large windows however suffer from the effect of perspective distortion and the robustness deteriorates. Based on the assumption that the geometric information from the 2D image features provides sufficient constraints for outlier detection a purely geometry-based outlier detection method was developed.

4.2.5.2 Geometry-based outlier removal

The following approach is designed to perform an estimation of the covariance of the extracted 3D primitives via error propagation. This mathematical framework allows to estimate an uncertainty measure for the parameters of the reconstructed primitives. Given an estimate of the a-priori uncertainty of the 2D primitives (edgels and ridgels in our case) the estimation of the final uncertainty (encoded by a covariance matrix) of the reconstructed primitives is performed using the error propagation principle. From this covariance matrices and the remaining residuals the uncertainties of the parameters of a 3D primitive that is reconstructed from several 2D image observations can be estimated. The distinction between inliers and outliers is then achieved by a simple thresholding on the uncertainty level of the individual parameters. In typical photogrammetric applications the uncertainties of all estimated parameters is computed during the non-linear bundle adjustment (see Luhmann et al. [122] pp. 234 ff.). The situation in the present case is somewhat easier since the 3D primitive is computed directly by a fast linear method and only the a posteriori covariance has to be computed.

The theoretical framework for error propagation is explained in detail in Luhmann et al. [122] page 52 ff. A compact outline that follows the notation of the book will be given below. The basis for the estimation of the uncertainties is the functional model for the least squares adjustment:

Given a vector **L** of observations:

$$\mathbf{L} = (L_1, L_2, \dots L_n)^T \ , \tag{4.1}$$

we want to determine the values for a vector $\mathbf{X}$ of unknown parameters:

$$\mathbf{X} = (X_1, X_2, \ldots X_u)^T \ , \tag{4.2}$$

where the number of observations n must be larger than the number of unknown parameters u. The functional model expresses the relation between the "true" observations $\tilde{\mathbf{X}}$ and the "true" unknowns $\tilde{\mathbf{L}}$ as a nonlinear correction equation:

$$\tilde{\mathbf{L}} = \varphi \tilde{\mathbf{X}} \ , \tag{4.3}$$

where φ is a vector of functions of the unknowns. Due to the fact that the true observation values are not known, $\tilde{\mathbf{L}}$ is replaced by the measured observations $\mathbf{L}$ and associated residual vector $\mathbf{v}$. Likewise the vector of unknowns $\tilde{\mathbf{X}}$ is replaced by the estimated unknowns $\hat{\mathbf{X}}$. This results in:

$$\mathbf{L} + \mathbf{v} = \varphi \hat{\mathbf{X}} \ . \tag{4.4}$$

For given approximate values $\mathbf{X^0}$ for the unknowns the estimated unknowns can be written as:

$$\hat{\mathbf{X}} = \mathbf{X^0} + \hat{\mathbf{x}} \ . \tag{4.5}$$

From $\mathbf{X^0}$, approximate values for the observations can be computed as

$$\mathbf{L^0} = \varphi(\mathbf{X^0}) \ . \tag{4.6}$$

The so called reduced observations are then computed as:

$$\mathbf{l} = \mathbf{L} - \mathbf{L^0} \ . \tag{4.7}$$

For small values of $\hat{\mathbf{x}}$ the correction equation is expanded into a first-order Taylor series and after introduction of the Jacobian matrix $\mathbf{A}$ the linearized correction equations are obtained as:

$$\hat{\mathbf{l}} = \mathbf{l} + \mathbf{v} = \mathbf{A}\hat{\mathbf{x}} \ . \tag{4.8}$$

The Jacobian matrix $\mathbf{A}$ contains the partial derivatives of the functions in φ with respect to the unknown parameters. Finally the update equation for the unknowns follows as:

$$\hat{\mathbf{x}} = (\mathbf{A^T A})^{-1} \mathbf{A^T l} \ . \tag{4.9}$$

The matrix

$$\mathbf{Q} = (\mathbf{A^T A})^{-1} \tag{4.10}$$

is called the cofactor matrix. From the residuals vector $\mathbf{v}$ for the adjusted parameters the *a posteriori* standard deviation can be expressed as:

$$\hat{s_0} = \sqrt{\frac{\mathbf{v^T P v}}{n-u}} \ . \tag{4.11}$$

The covariance matrix for the adjusted parameters follows as:

$$\mathbf{K} = \hat{s_0}^2 \mathbf{Q} \ . \tag{4.12}$$

In the present case we are interested in estimating the uncertainties of the parameters of the reconstructed 3D primitives. The primitives are represented a 3D position with an associated 3D direction vector, thus the parameters are equal to that of a 3D line. The minimal representation of a 3D line $\mathbf{G}$ uses four parameters [8], but we choose a representation that uses five parameters $\mathbf{G} = [g_1, g_2, \ldots g_5]$, but is more convenient for the purpose of estimating the covariance matrix of the parameters. The first three parameters encode the position in 3D space $\mathbf{p} = (g_1, g_2, g_3)$ and the last two parameters encode the lines normalized direction vector $\mathbf{a}$, using the two spherical angles: $\theta = g4 = \tan^{-1}(\frac{y}{x})$ and $\varphi = g5 = \cos^{-1}(z)$. Thus $\mathbf{a} = (\cos\theta \sin\varphi, \sin\theta \sin\varphi, \cos\varphi)$ and any point $\mathbf{p_i}$ on the line can be expressed as a function of the parameter t by: $\mathbf{p_i} = \mathbf{p} + \mathbf{a}\ t$.

For the purpose of estimating the uncertainty of the line parameters we can divide the problem. First we estimate the uncertainty of the location $\mathbf{p}$ and then the uncertainty of the spherical angles θ and φ.

For computing the Jacobian matrices the partial derivatives with respect to the parameters are computed numerically, using a finite differencing scheme. The Jacobian matrix $\mathbf{A_p}$ for the position is a $n \times 3$ matrix, where n is the number of 2D edgel measurements in the images. The residual vector $\mathbf{v_p}$ is the vector of perpendicular distances of the reprojected 3D position to the image measurements. As described in 4.2.4.2 the 3D position is the intersection of the 3D search ray that is defined by the a feature's position in the reference camera and the reconstructed 3D line. The image measurements are the corresponding 2D features in the images. The covariance matrix for the position results as:

$$\mathbf{K_p} = \hat{s_{p0}}^2 \mathbf{Q_p} = \sqrt{\frac{\mathbf{v_p^T P v_p}}{n-u}}\ (\mathbf{A_p^T A_p})^{-1}. \tag{4.13}$$

The Jacobian matrix $\mathbf{A_a}$ for the spherical angles is a $n \times 2$ matrix. The residual vector $\mathbf{v_a}$ is the vector of enclosed angles of the reprojected 3D direction with respect to the orientation vector of the 2D image measurements - the features tangent directions in this case. The tangent direction of the features is perpendicular to the features normal vector. The covariance for the orientation results as:

$$\mathbf{K_a} = \hat{s_{a0}}^2\mathbf{Q_a} = \sqrt{\frac{\mathbf{v_a^T P v_a}}{n-u}}\,(\mathbf{A_a^T A_a})^{-1}. \tag{4.14}$$

The weight matrix $\mathbf{P}$ encodes the a priori uncertainty of the 2D image measurements. For the experiments, typical values for the uncertainties are estimated from simulations on synthetic data.

By computing the eigenvalues and eigenvectors of the estimated covariance matrices the magnitudes and directions of the uncertainties can be computed. The eigenvalues of K_p, $[\lambda_{p1}, \lambda_{p2}, \lambda_{p3}]$ represent the variances of the 3D position, and the eigenvalues of K_a, $[\lambda_{a1}, \lambda_{a2}]$ are the variances of the spherical angles θ and φ. Since the matrices K_p and k_a are positive definite, all eigenvalues are also zero or positive. The eigenvalues are sorted in ascending order so that $\lambda_{p1} \leq \lambda_{p2} \leq \lambda_{p3}$, and $\lambda_{a1} \leq \lambda_{a2}$ holds. The standard deviations $[\sigma_{p1}, \sigma_{p2}, \sigma_{p3}]$ and $[\sigma_{a1}, \sigma_{a2}]$ are then computed as the square roots of the variances.

The standard deviations for the parameters of the 3D position $[\sigma_{p1}, \sigma_{p2}, \sigma_{p3}]$, that are computed from the eigenvalues of the covariance matrix K_p should have the following properties: The eigenvalue σ_{p_3} should be significantly larger than $max(\sigma_{p1}, \sigma_{p2})$ - this would be the eigenvalue that corresponds to the 3D primitives dominant eigenvector, that is the direction vector. The magnitude of other two eigenvalues depends on the residuals of the projected 3D position with respect to the 2D image measurements and the geometric configuration of the cameras (intersection angles of the 3D planes). For the standard deviations of the spherical angles no statement about general properties can be made.

Thus, for the outlier detection the magnitudes of $[\sigma_{p1}, \sigma_{p2}]$ and $[\sigma_{a1}, \sigma_{a2}]$ are of interest. The separation of inliers and outliers is achieved by two thresholds T_p and T_a, one for the position uncertainty (T_p) and one for the orientation uncertainty (T_a). A 3D hypothesis with all position uncertainties below the threshold: $max(\sigma_{p1}, \sigma_{p2}) < T_p$ and all angular uncertainties below the threshold: $max(\sigma_{a1}, \sigma_{a2}) < T_a$ is accepted as a valid hypothesis.

Up to now all outlier detection approaches only focussed on the primitives detected for a single 3D search ray. The fact that, for the proposed approach, every 2D feature also belongs to a contour chain allows to use this additional geometric information to devise robust outlier detection method based on energy minimization.

4.2.5.3 Outlier Removal based on Energy Minimization

The following method exploits the geometric properties of the extracted 3D hypotheses in order to perform a robust outlier detection. This approach basically assumes that the input features are grouped to form contour chains and furthermore that the 2D features that are grouped within a contour chain in 2D, also describe a smooth contour in 3D. Based on this assumption the selection of valid hypotheses is achieved by applying an energy minimization scheme. Basically, we are trying to match a deformable model to data points by means of energy minimization.

The input data for this optimization step are all 3D hypothesis that are detected for each 3D search ray. The contour linking/chaining in the reference image provides the ordering of the hypotheses in 3D. Figure 4.15 shows the principal configuration: For the search ray r_k multiple 3D hypotheses are detected (three in this illustration). The correct hypothesis (black dot) is not necessarily the one with the highest certainty. The incorrect hypotheses (marked as circles) are generated by primitives from contours that also fulfill the multi-ray convergence criterion.

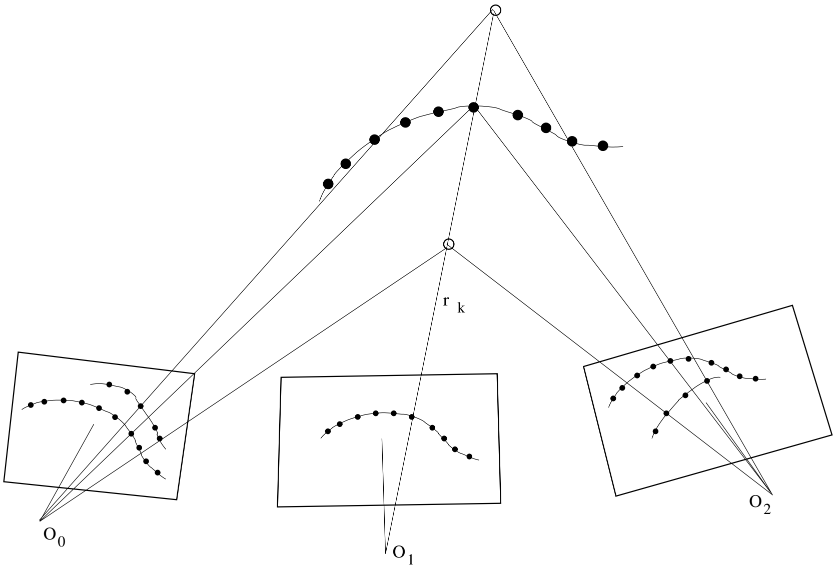


Figure 4.15: Multiple 3D hypotheses on a 3D search ray r_k. During the sweeping process multiple hypotheses per 3D search ray r_k can be encountered. In this illustration three hypotheses were detected - the correct hypothesis (marked by a dot) and two incorrect hypotheses (marked as circles) that are generated by features from image contours that also fulfill the multi ray convergence criterion.

The proposed approach takes all 3D hypotheses that are detected for a single 2D contour in the reference image into account. Figure 4.16 illustrates an example of such an optimization scenario using six rays $r_0 \dots r_6$. For every individual ray all valid depth hypotheses (those with a sufficient support count, and low uncertainty) are stored (e.g. $p_{0,0} \dots p_{0,3}$ for r_0). The aim of the optimization is now to detect the optimal contour path (shown as continuous line) within the upper and lower envelope (shown as dotted lines).

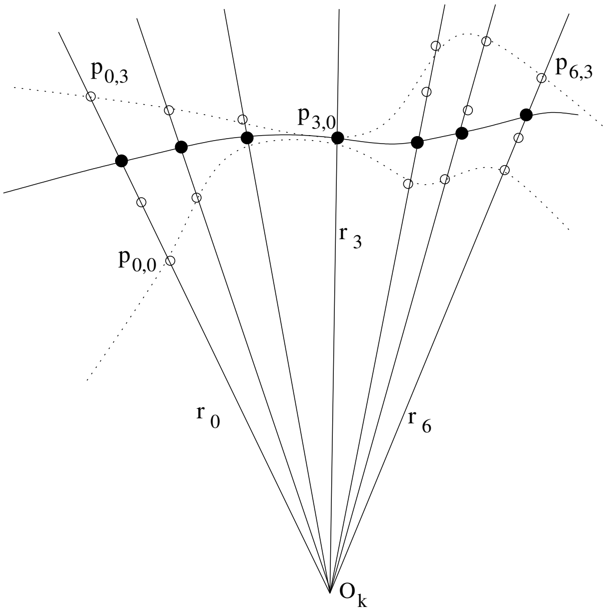


Figure 4.16: Illustration of the input data for the optimization step: For every search ray $r_0 \dots r_6$ all detected hypotheses (e.g. $p_{0,0} \dots p_{0,3}$ for r_0) are stored. The dashed lines show the upper and lower envelope of the possible contour paths, whereas the continuous line shows the correct contour path.

The proposed approach makes several simplifications and should only be considered as a proof of concept. Basically the problem of selecting the inlier points is based on the following assumptions:

1. The smoothness of the 3D contour is measured by the depth values of the 3D primitives of the contour. The depth of a primitive is measured from the origin of the reference camera (O_k in figure 4.16).

2. The number of outliers is significantly smaller then the number of correct 3D primitives.

3. Only the 3D position of the primitives is considered in the optimization process, the orientation is neglected.

The simplified assumption that only depth values are used, allows to formulate the optimization as a 2D problem. Therefore the implicit function that optimally approximates the 3D primitives while adhering a smoothness constraint is actually found in 2D. Figure 4.17 illustrates how the example shown in figure 4.16 is transferred to a 2D problem.

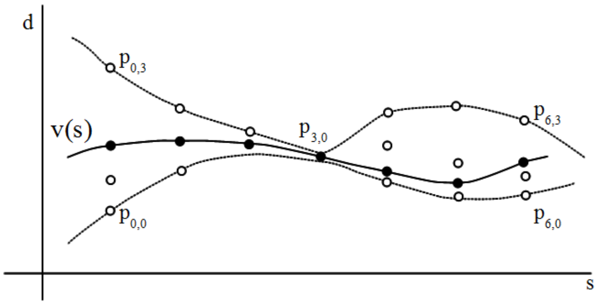


Figure 4.17: Illustration of the 2D energy minimization problem. On the y-axis the distance d of the 3D primitives $p_{i,j}$ to the reference camera origin (O_k in figure 4.16) is shown and the x-axis denotes the parameter s that varies from [0, 6] along the length of the contour in the reference image. The dashed lines show the upper and lower envelope of the possible contour paths, whereas the continuous line shows the correct contour path.

The extraction of the optimal path involves the solution of an energy minimization problem. Given the two main conditions for the minimization:

1. The resulting path should connect the 3D primitives and minimize the distance to the primitives.

2. The resulting path should be smooth.

The first condition is the so called data fidelity and the second condition is a smoothness constraint. A similar problem is solved for the active contour models that are also called *snakes*. Introduced by Kass et al. [66]) the energy functional for an active contour model $v(s)$ is defined as:

$$E_{total} = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s)). \tag{4.15}$$

E_{int} is the internal energy of the contour model and depends solely on its shape. E_{image} denotes the image energy and depends on the image intensity values along the path of the contour model and E_{con} is the constraint energy that can be based on artificial energy fields imposed by the user.

In our case E_{image} is renamed to E_{data} and measures the distance of the contour model to the data points (the converted primitives $p_{i,j}$ in 2D space as shown in figure 4.17). Thus, E_{data} measures the data fidelity. This is the difference between the measured depth values and the current value of the functional. In order to be robust against outliers the data residuals are computed using the Huber kernel. Based on a specified threshold σ Huber kernel weighs the distance $r(s) = \mathbf{v}(s) - p_{s,i}$ between the functional and the depth of a hypothesis as:

$$\rho(r(s), \sigma) = \begin{cases} r(s)^2 & \text{if } r(s) \leq \sigma \\ 2\sigma|r(s)| - \sigma^2 & \text{if } r(s) > \sigma \end{cases} \tag{4.16}$$

Since for one particular depth value several measurements may exist a single data residual is expressed as:

$$E_{data}(s) = \frac{1}{i} \sum_{i=1}^{n} \rho((\mathbf{v}(s) - p_{s,i}), \sigma) \ . \tag{4.17}$$

Where $\mathbf{v}s$ is the current value of the functional for the ray r_s and $p_{s,i}$ is the depth of the primitive $p_{s,i}$

E_{int} measures the local smoothness using the second order derivative:

$$E_{int} = \left(\frac{\partial^2 \mathbf{v}(s)}{\partial s^2} \right)^2 \ . \tag{4.18}$$

As proposed in [66] the derivative is approximated by finite differences:

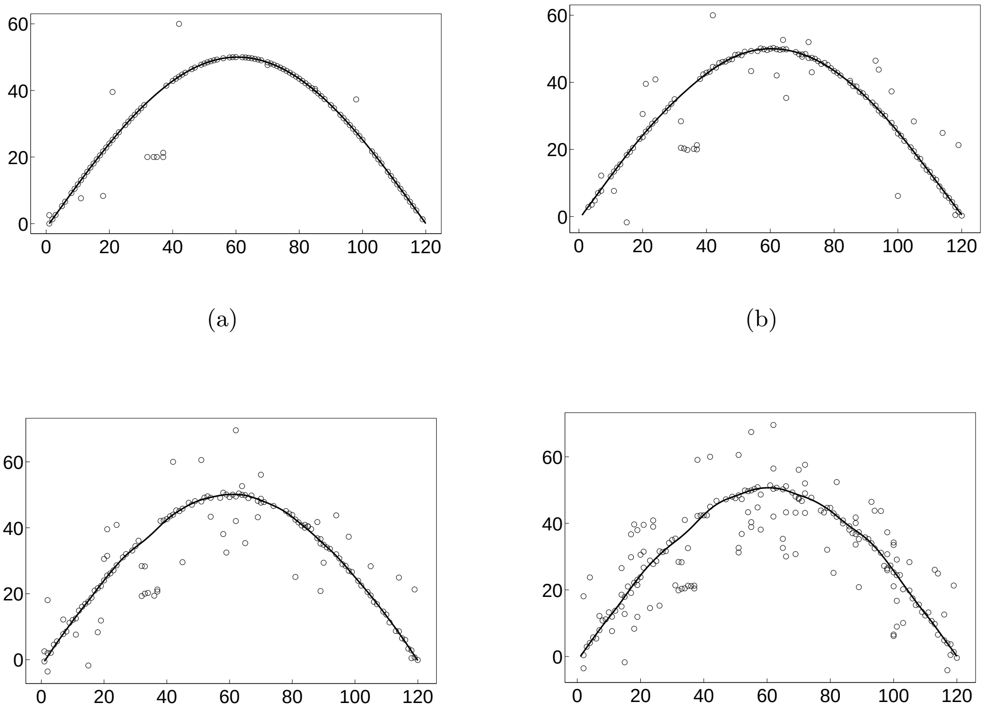
$$\frac{\delta^2 \mathbf{v}(s)}{s^2} \approx \mathbf{v}(s+1) - 2\mathbf{v}(s) + \mathbf{v}(s-1) \ . \tag{4.19}$$

The third term of the energy integral comes from external constraints imposed either by user interaction or some higher level interpretation. This term is neglected in our case.

This energy formulation is then minimized using a variant of the Levenberg-Marquardt algorithm. The Jacobian matrix A that holds the partial derivatives of the unknowns (see 4.9) is also computed using finite differences. Since the smoothness term at a particular position $\mathbf{v}(s)$ is only dependent on is immediate predecessor $\mathbf{v}(s-1)$ and successor $\mathbf{v}(s+1)$ the Jacobian matrix A is a tridiagonal banded matrix and the matrix product $A^T A$ is a pentadiagonal banded matrix. The inverse of a non-singular pentadiagonal matrix can be computed efficiently as proposed in [32] pp.98 ff.

Using this global optimization method the number of outliers could be reduced by 83%

in experiments with synthetic data. In figure 4.18 examples for the performance of the energy minimization approach on synthetic data are shown. The true hypotheses lie on the upper half of a sine wave that is scaled by a factor of 50.0. The number of true points is 120, but 20 points are removed at random positions and from position 30 to 35 a series of outliers is simulated, making the number of true hypotheses 94. The number of outlier points increases from 10 to 30, 50 and 100. The noise level (uniform random noise is added to the y-coordinates of the points) increases from 0.0 to 0.5, 1.0 and 2.0. The functional $\mathbf{v}(s)$ is initialized with the average depth from all hypotheses - this make the initial shape a horizontal line.



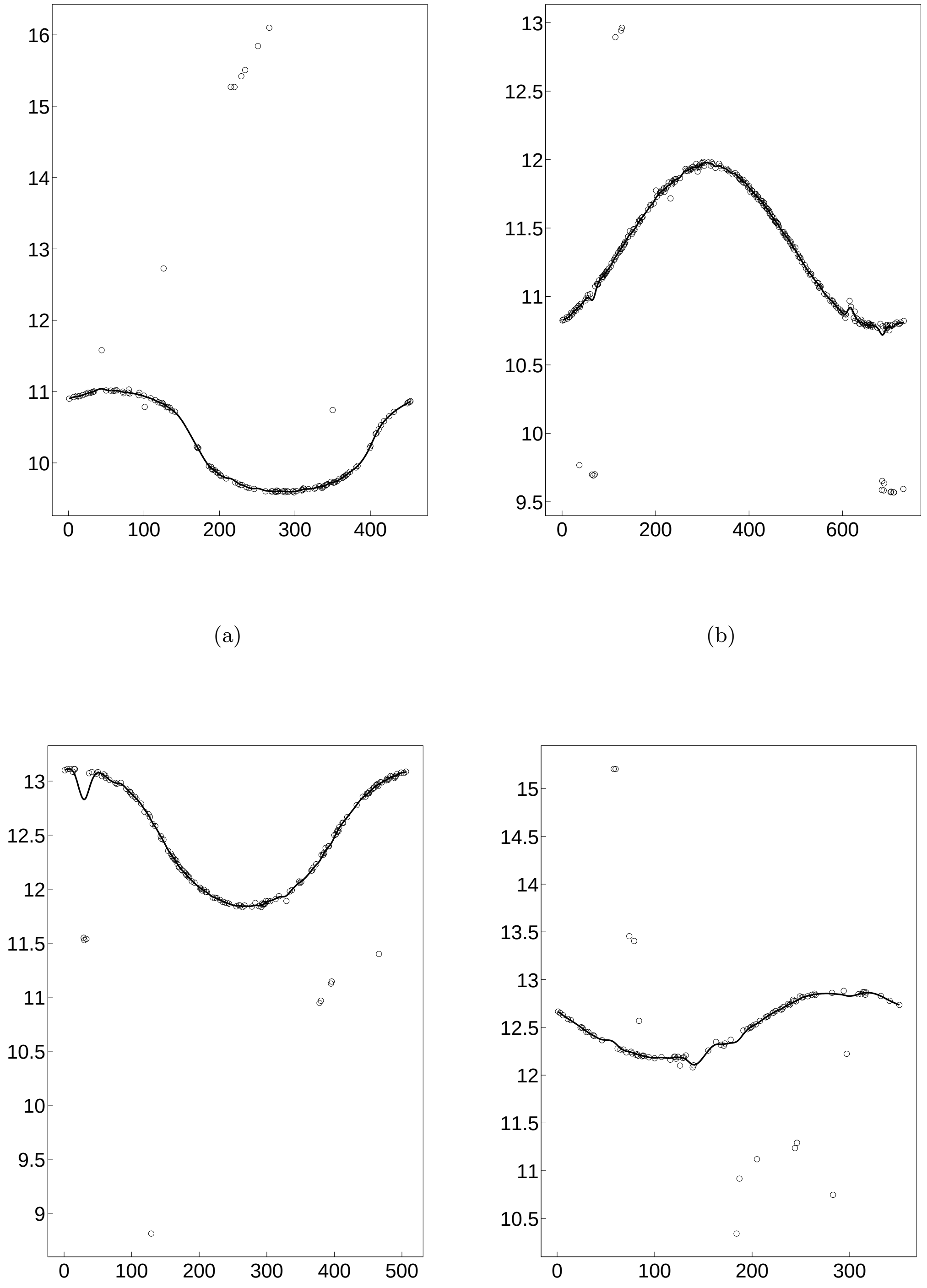
(a)

(b)

(c)

(d)

Figure 4.18: Examples for outlier detection by energy minimization on synthetic data. The true points lie on the upper half of a sine wave (scaled by a factor of 50.0). The number of true points is 120, but 20 points are removed at random positions and from position 30 to 35 a series of outliers is simulated, making the number of true hypotheses 94. The number of outlier points increases from 10 in (a) to 30 in (b), 50 in (c) and 100 in (d). The noise level (uniform random noise is added to the y-coordinates of the points) increases from 0.0 in (a) to 0.5 in (b), 1.0 in (c) and 2.0 in (d). The functional $\mathbf{v}(s)$ is initialized with the average depth from all hypotheses - this make the initial shape a horizontal line.



(a)

(b)

(c)

(d)

Figure 4.19: Examples for outlier detection by energy minimization on real data. The plots (a) to (c) illustrate the performance of the energy minimization method on real data. In typical data sets the number of outliers is significantly lower than the number of correct hypotheses. As in the examples with synthetic data the functional $\mathbf{v}(s)$ is initialized with the average depth from all hypotheses.

4.2.6 Concluding Remarks

The presented method allows for the efficient computation of sparse clouds of 3d primitives (3D points with associated direction vector) from 2D features that are extracted in multiple oriented images. The main contribution is the use of stereo rectification to accelerate the spatial query for edgel candidates the lie close to the epipolar line, the incorporation of directional information associated to the 2D features in the selection process and the event driven 3D hypothesis generation using an uncertainty measure. The image-based outlier detection proved not to be sufficient for the robust detection of outliers. However, the energy minimization approach shows promising results and can be used to achieve robust verification of 3D hypotheses. This introduction of a global optimization scheme carried out on contours significantly increased the robustness of the process.

Chapter 5

Discussion based on data

Contents

The analysis of the reconstruction accuracy of an image-based modeling framework includes the study of influences from different sources of error. For measurements taken with a camera the main factors that influence the accuracy of the reconstructed 3D points are the accuracy of the measurements taken from the images, the accuracy of the image orientation process, the accuracy of the intrinsic camera parameters (focal length, principal point, lens distortion parameters) and the geometric configuration, also known as strength, of the camera positions and orientation. Furthermore the overall complexity of the observed scene influences the

The experimental section is structured as follows:

1. Analysis of the extraction accuracy of edges and ridges in individual images. The findings provide estimates for all subsequent extraction/modeling methods.

2. Analysis of the extraction accuracy of straight 2D lines segments and vanishing points in single images.

3. Evaluation of the performance of model fitting. In this section the ellipse fitting and the affine square fitting methods are evaluated, again on single images.

4. Assessment of the matching performance of the descriptors from section 3.2.

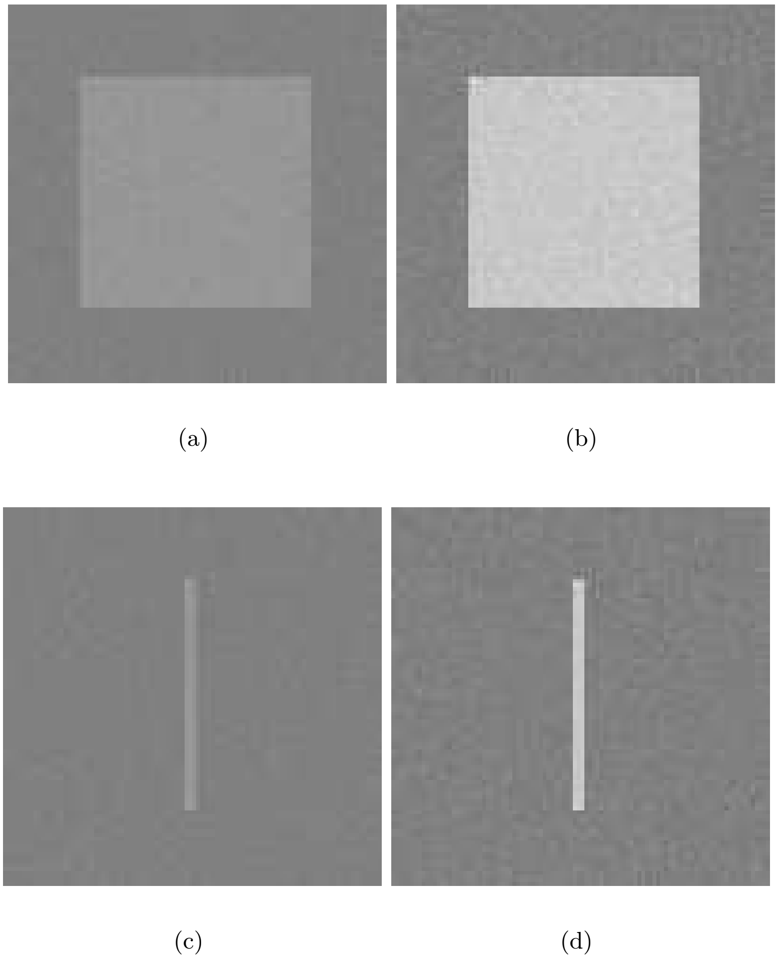
5. Analysis of the reconstruction accuracy of the space sweeping approach.

5.1 Evaluation framework

For the evaluation of the extraction accuracy of image features we use a framework that generates synthetic images. The analysis of feature extraction methods using synthetic images has the advantage that the ground truth is known. In order to avoid sampling artifacts, that get introduced when primitives with non axis aligned lines or curved outlines are rendered in an image, only simple geometric primitives consisting of axis aligned straight line segments are used. In the present case we use axis aligned squares and lines. The images have a radiometric resolution of 8bit (256 intensity values). Figure 5.1 shows two rows of example images: The first row contains images of an axis aligned square filled with different shades of gray. The square is placed on a mid gray back ground. These images are used to test the accuracy of edgel extraction. The second row shows images of a vertical line, again with different shades of gray, that is placed on a mid gray back ground. These images are used for assessing the accuracy of the ridge extraction method. To all test images Gaussian noise is added to the intensity values of the image pixels. The standard deviation σ in the depicted images is 3 and 5 intensity values. Subsequently the feature extraction is performed on these images. Finally, the evaluation the positions and orientations of the extracted features are compared to the ground truth data.

The performance analysis for the vanishing point extraction methods is also carried out on synthetic data. These synthetic data sets are created by defining a set of random vanishing points $p_0 \dots p_n$ in the bounded 2D plane ($-10^5 < x < 10^5$, $-10^5 < y < 10^5$). The number of vanishing points is chosen to correspond to typical urban scenes which exhibit one to five dominant vanishing points. The set of supporting 2D line segments is generated in a bounded image plane. The number of lines supporting a particular vanishing point is randomly varied between predefined bounds (30 to 1500 in our experiments). In order to simulate real world conditions, the length of the line segments varies from 20 to 300 pixels and the direction vector of the 2D lines is corrupted with increasing amounts of Gaussian noise ($\sigma = 1..5^o$). Furthermore a variable number of random line segments is inserted to test the robustness of extraction. Figure 5.2 shows an illustration of the synthetic setup:

The evaluation of the performance of fitting affinely distorted squares to point sets is



(a) (b)

(c) (d)

Figure 5.1: Illustration of synthetic test images for the feature extraction experiments. Top row: Two images with an axis aligned square - these images are used to evaluate the extraction accuracy of edgels and straight 2D lines. Bottom row: Two images with a vertical line - these images are used to evaluate the extraction accuracy of ridgels. The primitives in the images have increasing contrast (left: 20 intensity values, right 70 intensity values) and the images are contaminated by increasing amounts of additive Gaussian noise (radiometric resolution = 8bit, image noise left $\sigma = 3.0$, image noise right $\sigma = 5.0$).

also performed on synthetic data. In this experiment the 2D points of a synthetic square are transformed by a random affine transform. In order to assess the robustness of the method, the point coordinates are perturbed by increasing amounts of Gaussian noise. Effects of imperfect segmentation are introduced by disturbing portions of the square points by a smoothly varying random function. Thus the square fitting method faces

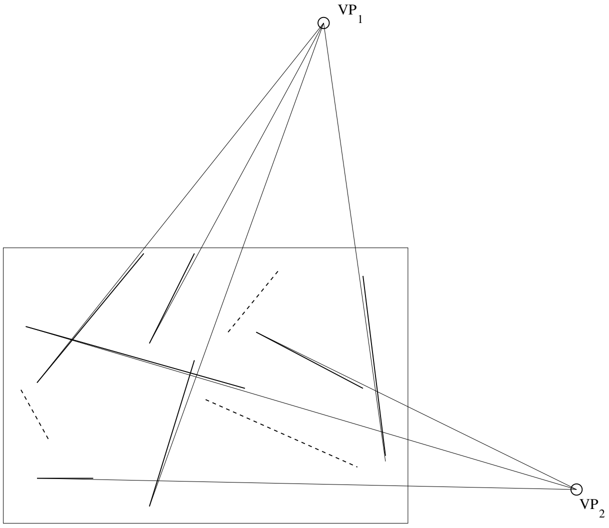
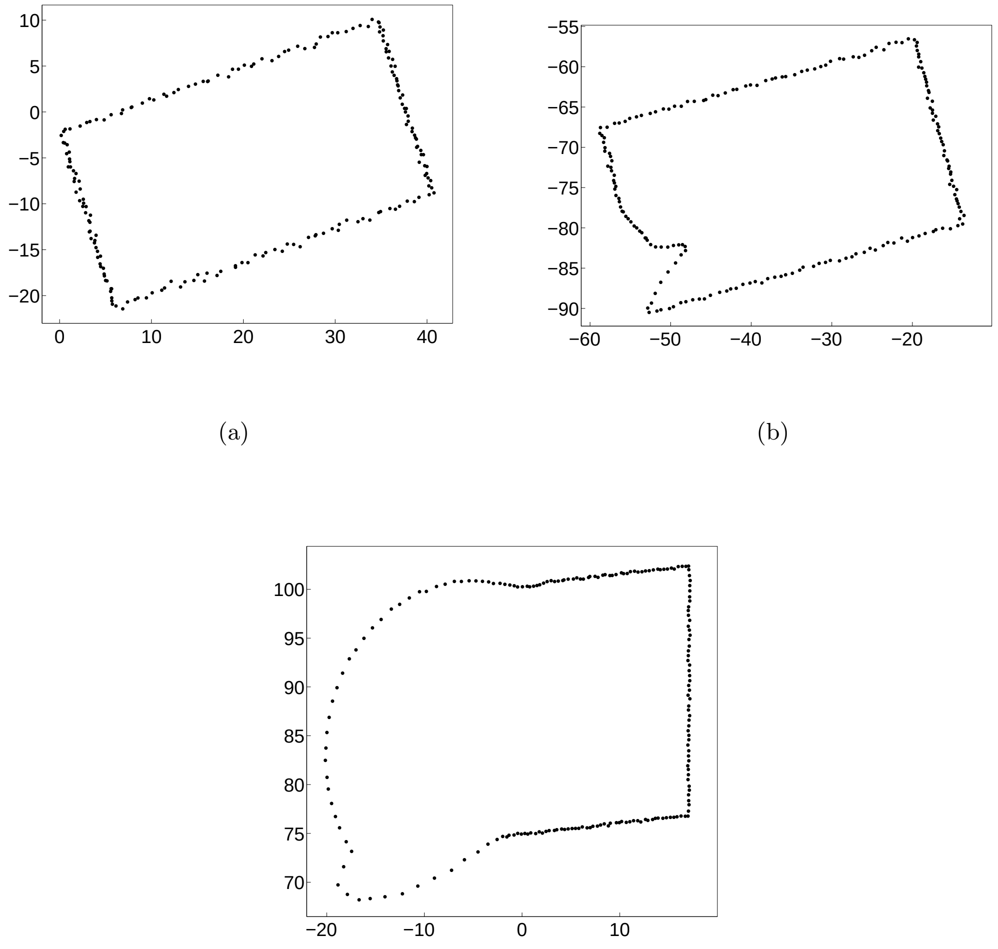


Figure 5.2: Illustration of a synthetic line set for the analysis of the vanishing point extraction methods. A number of line segments supports two vanishing points (VP_1 and VP_2 shown as circle). The orientation of the line segments deviates from the true orientation shown as thin continuous lines. Additionally a number of line segments that do not support any vanishing point is present (shown as dashed lines). Note that the indicated image frame is for illustration purposes only - the features are not rendered into a synthetic image, but are directly fed to the vanishing point extraction methods.

the challenge of determining the affine transform parameters from a noisy point set that contains outliers. In figure 5.3 examples for the test data sets are shown: From left to right the amount of Gaussian noise as well as the number of outlier points increases. The result of the fitting process is an affine transform that should match the generating transform as close as possible.



(a)

(b)

(c)

Figure 5.3: Illustration of synthetic 2D point sets for the evaluation of the affine square fitting. From left to right the number of outlier points increases (left=0%, middle=15%, right=30%) and the point locations are perturbed by decreasing amounts of additive Gaussian noise (left $\sigma = 0.21$, middle $\sigma = 0.15$, right $\sigma = 0.06$). Note that the illustrated feature points are not rendered into a synthetic image, but are directly fed to the affine square fitting method.

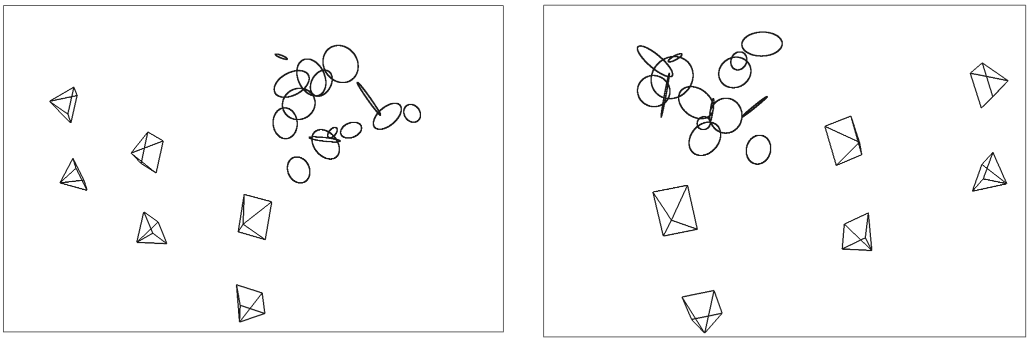
The evaluation of the multi-view space-sweeping method for the generation of sparse point clouds from directed 2D primitives (edgels and ridgels) is also evaluated with synthetic data. The use of a synthetic data set for the assessment of the accuracy of the proposed space sweeping method allows to isolate the effects of individual parameters. With this synthetic setup it is for example possible to analyze the influence of errors in the image measurements without the effect of noisy camera parameters (intrinsic as well as extrinsic). For this purpose a dataset with noise-free camera parameters and noisy image features, in the particular case edges, is created.

Initially a set of synthetic cameras is generated, in the present case the cameras are all looking into the same general direction. An axis aligned bounding box that is fully visible for all cameras is defined - all 3D features are restricted to be in this box. This strategy ensures that a valid image projection for each 3D primitive exists. The scene features are 3D circles at a random position, with a random normal vector and a random radius. These circles have an analytic representation and are therefore well suited for evaluation purposes. The rounded outline of the circles allows to demonstrate the advantages of the proposed method over other feature-based modeling methods that operate on straight line segments.

From the 3D circles point sets are generated by sampling the circle at regular intervals. The spatial resolution is chosen so that the distance between two adjacent points in the image space of each camera is significantly smaller than one pixel - this ensures a contiguous chain when projecting into the image space of the synthetic cameras. The 2D image edges are directly created by re-projection of the synthetic 3D points instead of being extracted from images. The vector between adjacent projected points is used to compute a direction vector for these synthetic edges. Due to the high density of 3D points many points project onto one discrete pixel position and a subsequent thinning process ensures that only one edgel per pixel remains in the image. This ensures that circles that are farther from a particular camera are sampled at a physically correct rate. The effect of image noise is simulated by perturbing the position and the normal vector of the synthetic edgels with Gaussian noise.

With this data set it is now possible to independently vary parameters and observe the effects. Due to the known ground truth the evaluation is carried out in a straightforward way by simply comparing the reconstructed 3D primitives with their ground truth neighbor.

Figure 5.4 shows three views of a test scene with fifteen 3D circles that are observed by six cameras.



(a) (b)

Figure 5.4: Two renderings of a set of fifteen synthetic 3D circles for the evaluation of the space sweeping. The scene is observed by six synthetic cameras (symbolized by their frustums) and all circles are fully visible in all cameras. The synthetic edgels are created by projecting the 3D circle points into the cameras.

Having introduced the evaluation framework we now report the results of the experiments.

5.2 Accuracy of edge and ridge extraction

In this section the geometric accuracy of the various point-like image features is analyzed. This analysis is necessary since the geometric uncertainty of the extracted features influences all subsequent processing steps such as line extraction, vanishing point estimation, image orientation and 3D modeling. Geometric uncertainties are the deviations of the location of an extracted primitive from its true position. The degree of deviation is influenced by image noise and the types of geometric models that are used to estimate the location. All features that are investigated in this section are extracted with sub-pixel accuracy i.e. their position is not fixed to the integer grid of the image. The analysis starts with an assessment of the location of edges and ridges. The extraction methods are tested on synthetic and real images.

The synthetic test image for assessing the detection accuracy of the edge extraction method consists of a set squares. In order to avoid a bias from the rendering methods only axis aligned squares are used. The squares are rendered with decreasing contrast. For the experiment different amounts of Gaussian noise are added to the images. For every test image the edgels are extracted using the method described in section 2 - Canny's edge extraction approach plus estimation of the location estimation. The evaluation analyses the position and orientation of the extracted edges to the ground truth. Figure 5.5 shows a crop out of a test image with extracted edgels (red arrows) and ground truth edge (cyan line) overlaid.

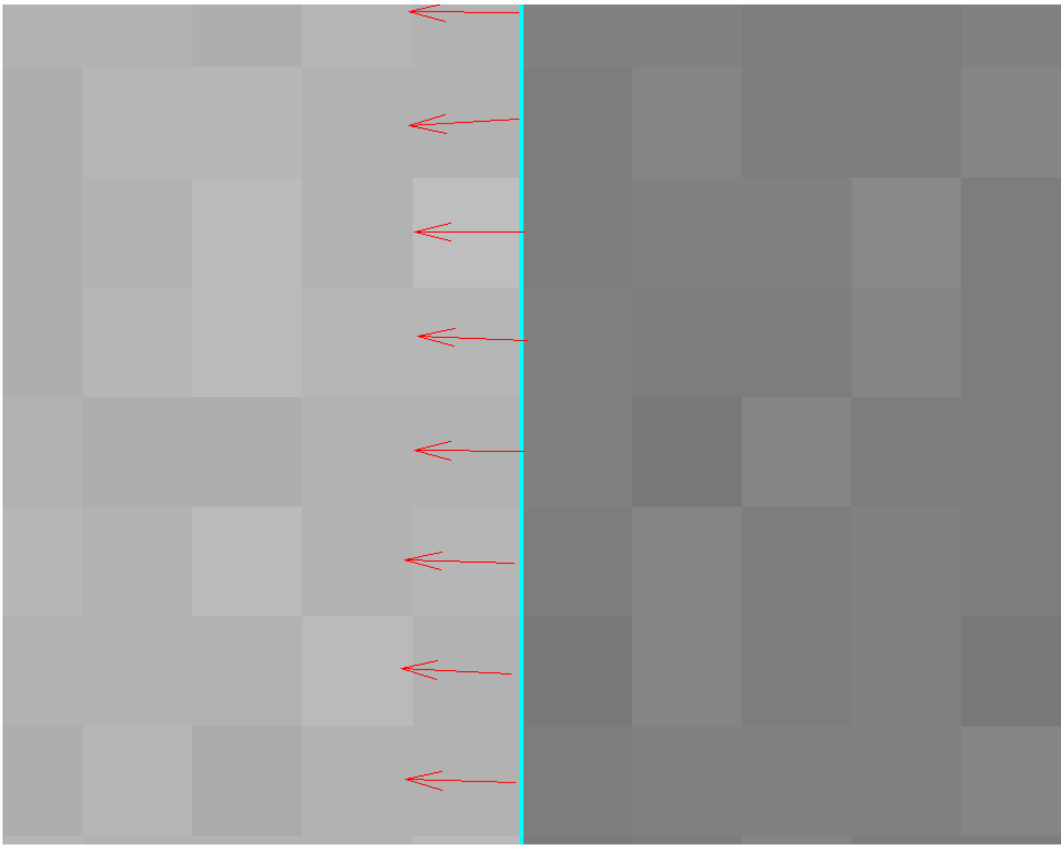


Figure 5.5: Crop out of edgel extraction result in a noisy test image. The extracted edgels are illustrated as red arrows and the ground truth as cyan line. Gaussian noise with a standard deviation of 3.6 gray levels (the full range is 256 levels) was added in this particular case.

Figure 5.6 lists the results of the edgel extraction on the synthetic images.

The intensity values of the 8-bit test images are corrupted by increasing amounts of additive Gaussian noise ($\sigma = 0.1 \ldots 5$) and three different contrast levels c are investigated namely 20, 50 and 100 intensity values - this corresponds to 7.8 percent, 19.5 percent and 39 percent of the intensity range of an 8bit image. Due to the known ground truth the evaluation of the extraction accuracy can by achieved in a straightforward manner. The edges that are extracted close to the corners of the square are neglected in order to avoid

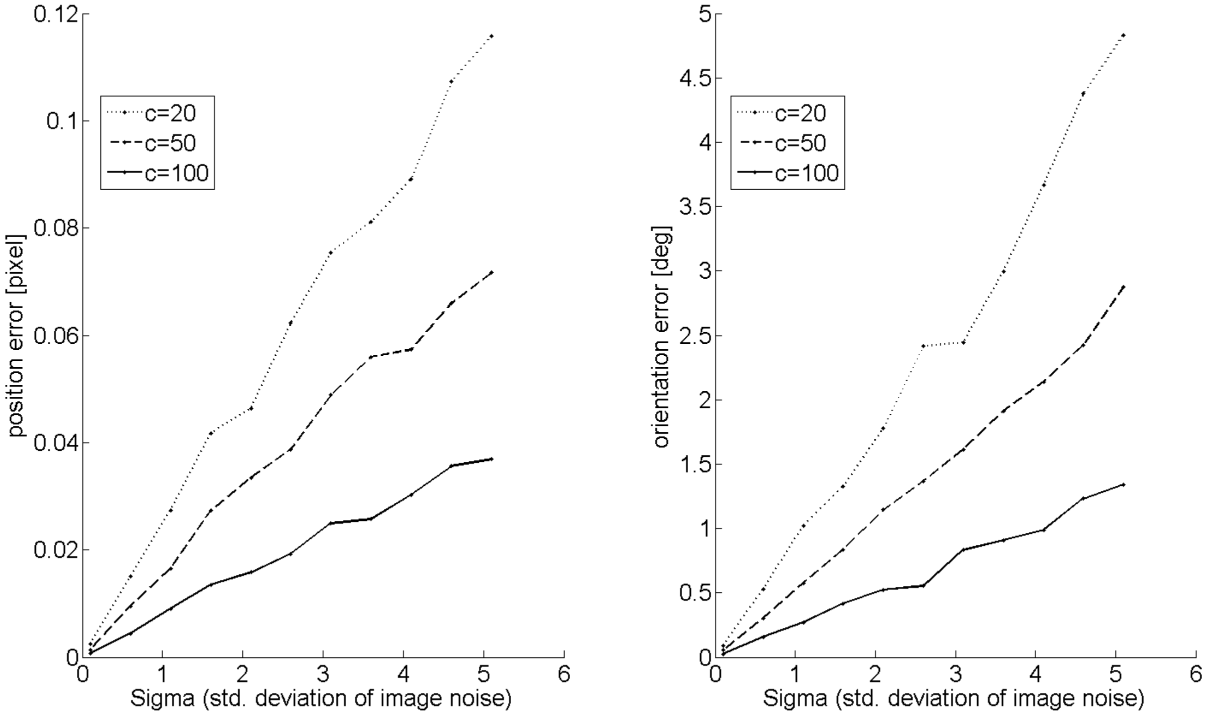


Figure 5.6: Results for the edgel extraction using Canny's extraction approach plus estimation of the location estimation. In order to simulate realistic imaging conditions the intensity values of the 8-bit test images are corrupted by increasing amounts of additive Gaussian noise ($\sigma = 0.1 \dots 5$). Three different contrast levels are investigated namely 20, 50 and 100 intensity values - this corresponds to 7.8 percent, 19.5 percent and 39 percent of the intensity range of an 8bit image. Due to the known ground truth the evaluation of the extraction accuracy can be achieved in a straightforward manner. The figure on the left shows the position error - which is the perpendicular distance of an edgel's position to the ground truth edge of the square. The figure on the right shows the orientation error - which is the difference angle between the estimated normal vector of an edgel and the ground truth normal vector of the square's edge. The main observation is that the position error remains low even under considerable noise and low contrast (maximum is 0.15 pixel for a contrast of 20 and $\sigma = 4.6$) whereas the orientation error is more sensitive to noise (maximum is 7.2 degrees for a contrast of 20 and $\sigma = 4.6$).

influences of The figure on the left shows the position error - which is the perpendicular distance of an edgel's position to the ground truth edge of the square. The figure on the right shows the orientation error - which is the difference angle between the estimated normal vector of an edgel and the ground truth normal vector of the square's edge. The main observation is that the position error remains low even under considerable noise and low contrast (maximum is 0.15 pixel for a contrast of 20 and $\sigma = 4.6$) whereas the orientation error is more sensitive to noise (maximum is 7.2 degrees for a contrast of 20 and $\sigma = 4.6$).

For testing the accuracy of ridge extraction the synthetic test images are filled with vertical lines of 3 pixel width (this is a typical width that also occurs in natural images). The background of the test image is mid gray and the lines are rendered with varying contrast levels - the same values as for the edgel extraction experiments (20, 50 and 100 intensity values).

Figure 5.7 shows a crop out of a test image with extracted ridgels (red arrows) and ground truth center line (in cyan) overlaid.

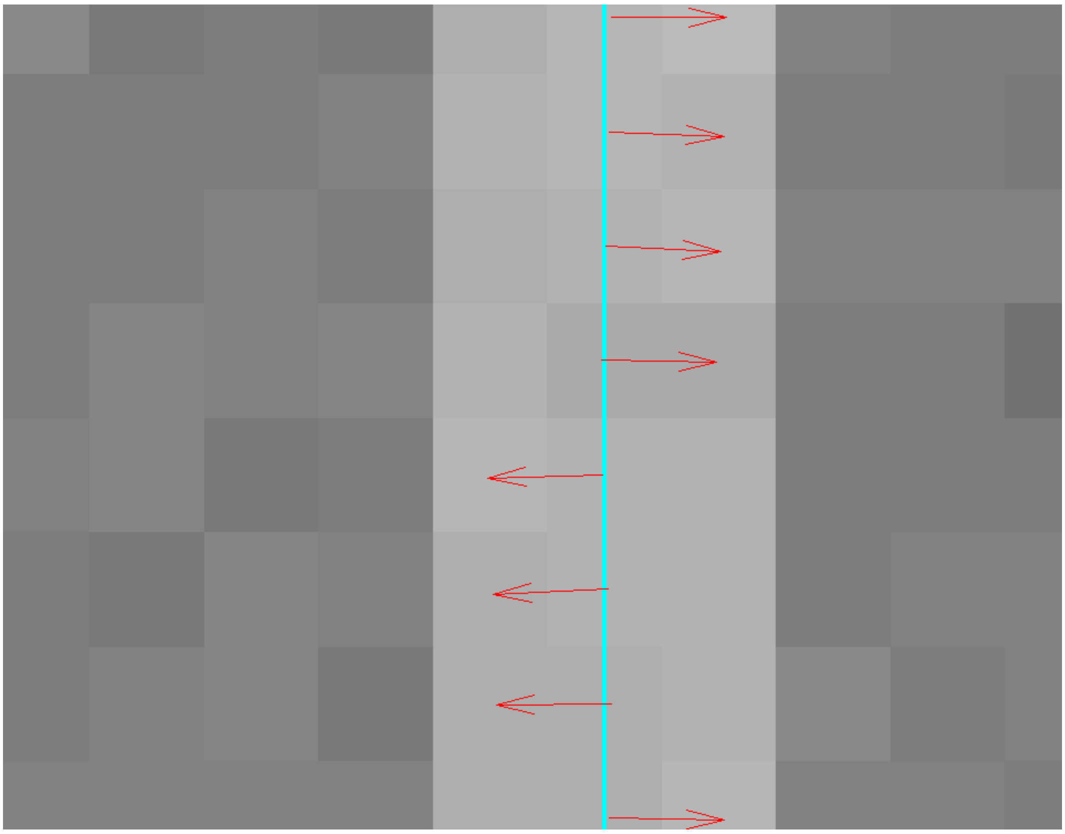


Figure 5.7: Crop out of ridgel extraction result in a noisy test image. The extracted ridgels are illustrated as red arrows and the ground truth as cyan line. Gaussian noise with a standard deviation of 3.6 gray values (the full range is 256 levels) was added in this particular case. Note that the orientation of the extracted ridges has an ambiguity of $\pm\pi$.

Figure 5.8 lists the results of the ridge extraction on the synthetic images.

The experiments in this section showed that directed features - in the particular case edgels and ridgels can be extracted with sub-pixel accuracy. For moderate image noise

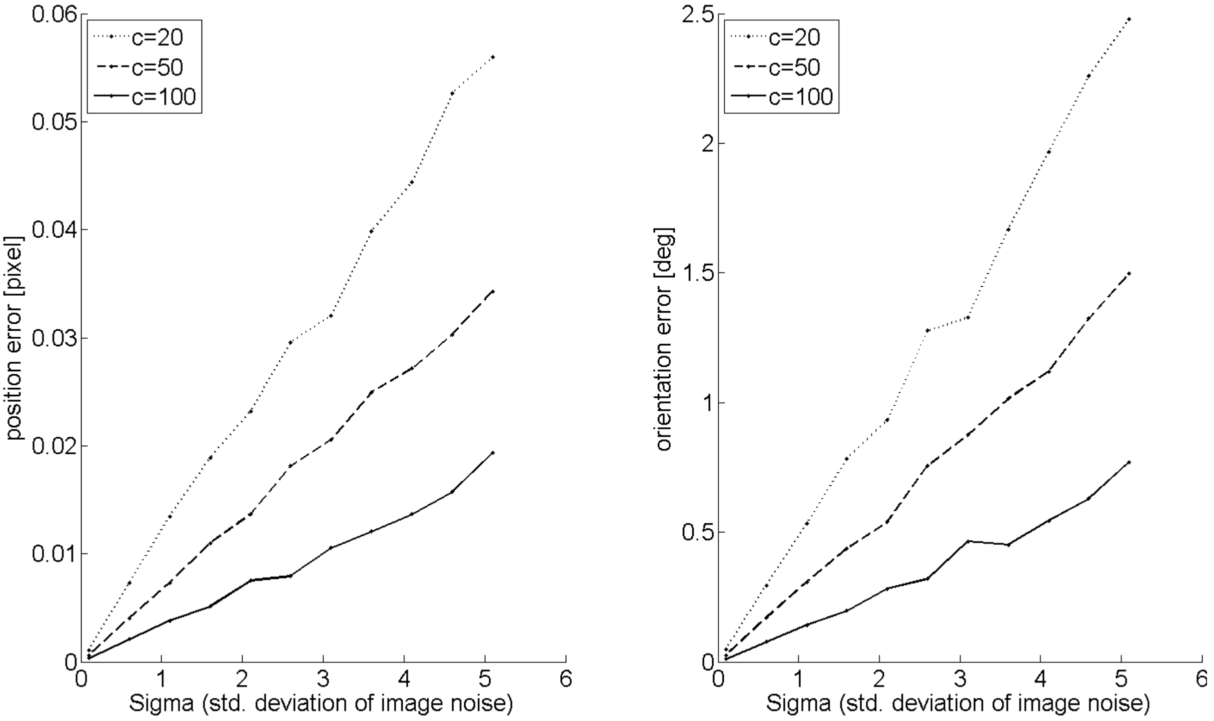


Figure 5.8: Results for the ridgel extraction. In order to simulate realistic imaging conditions the intensity values of the 8-bit test images are corrupted by increasing amounts of additive Gaussian noise ($\sigma = 0.1 \ldots 5$). Three different contrast levels are investigated namely 20, 50 and 100 intensity values - this corresponds to 7.8 percent, 19.5 percent and 39 percent of the intensity range of an 8bit image. Due to the known ground truth the evaluation of the extraction accuracy can by achieved in a straightforward manner. The figure on the left shows the position error - which is the perpendicular distance of a ridgel's position to the straight center line of the ridge. The figure on the right shows the orientation error - which is the difference angle between the estimated normal vector of a ridgel and the ground truth normal vector of the ridge's center line. The main observation is that the position error remains low even under considerable noise and low contrast (maximum is 0.15 pixel for a contrast of 20 and $\sigma = 4.6 graylevels$) whereas the orientation error is more sensitive to noise (maximum is 7.2 degrees for a contrast of 20 and $\sigma = 4.6 graylevels$). Another fact worth mentioning is the observation, that the location and orientation errors of ridges are significantly lower than the errors for edges.

($\sigma = 1 \ldots 5$), as it is typical for digital images that are captured with consumer cameras, the position accuracy is below 0.1 pixel and the orientation accuracy is below 5 degrees. The influence of image noise grows if the contrast between 'background' and 'foreground' decreases. This dependency should be taken into account when features are used for detecting more complex geometric primitives such as straight line segments or ellipses. In typical images the number of edgels is significantly higher than the number of ridgels but as the experiments showed: Ridgels are a class of directed primitives that can be extracted with an accuracy that is better than the achievable accuracy for edgels. In general ridges may serve to provide information in regions where edge detection fails or becomes prone to gross errors e.g. in cluttered regions.

5.3 Accuracy of 2D line segment and vanishing point extraction

For testing the accuracy of the 2D line segment extraction the same synthetic images as for the evaluation of the edgel extraction accuracy are used. For the Canny edges that are extracted in these images straight line segments are extracted. The estimated line parameters are then compared with the known ground truth, namely the sides of the squares. Figure 5.9 shows the mean error for the translational and the rotational part. The translational error is the average perpendicular distance of the two corresponding corners of the synthetic squares to the estimated line segment. The rotational error is the absolute enclosed angle between the normalized line vector of the estimated line and the normal vector of the ground truth edge of the square.

The most notable observation is that the accuracy of the 2D line segments is significantly higher than the accuracy of single edges. However keeping in mind that the estimation of the line parameters is performed by an outlier sensitive linear least squares approach, these results can only be expected when a proper inlier selection process, such as RANSAC, is performed in advance. In conclusion the detection and fitting of straight 2D line segments increases the geometric accuracy and provides a compact representation.

The following experiments will analyze the performance and geometric accuracy of Rother's vanishing point extraction (RM) and for the Thales circle method (TCM) that were both presented in Section 2.7. The first set of experiments uses synthetically generated line sets to assess the geometric accuracy.

Figure 5.10 shows the results for the two methods. The relative location error is the measured distance of the located vanishing point from the images principal point normalized by the true distance.

The main observation of this experiment is that the relative position error for Rother's method is approximately half as high as for the Thales circle method. However both methods perform well, even under the presence of a high number of outlier lines. The

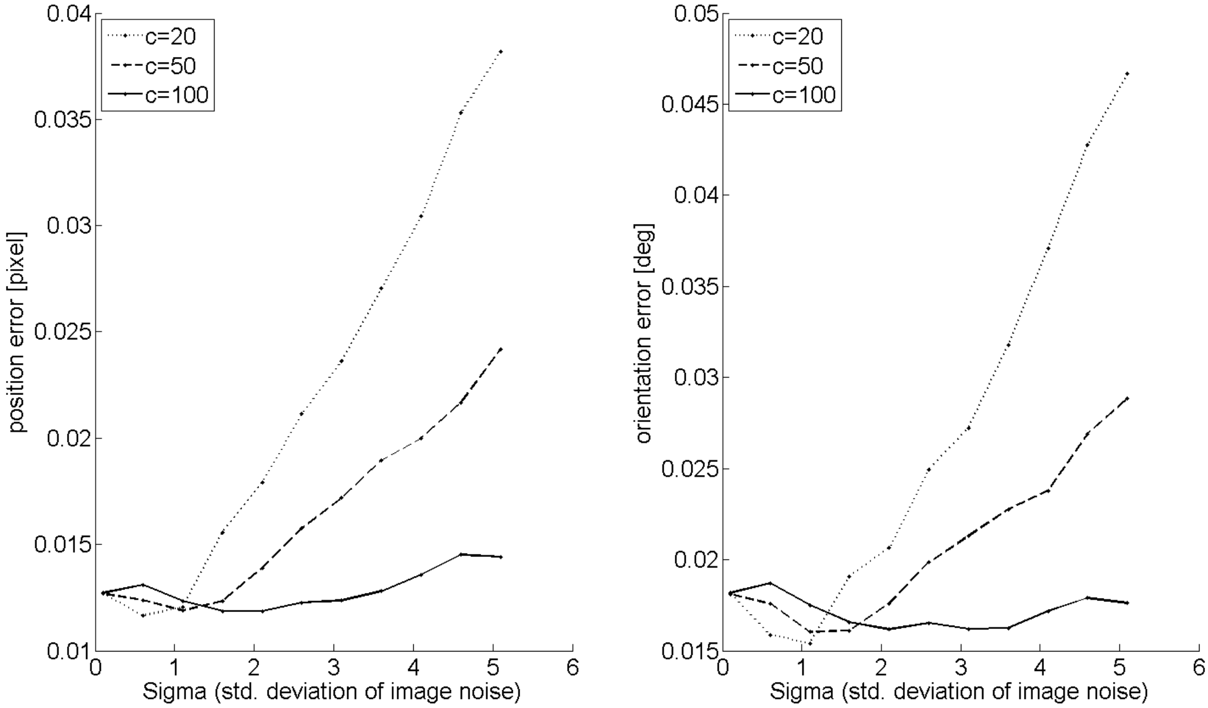


Figure 5.9: Results for the straight line extraction. In order to simulate realistic imaging conditions the intensity values of the 8-bit test images are corrupted by increasing amounts of additive Gaussian noise ($\sigma = 0.1 \dots 5$). Three different contrast levels are investigated namely 20, 50 and 100 intensity values - this corresponds to 7.8 percent, 19.5 percent and 39 percent of the intensity range of an 8bit image. Due to the known ground truth the evaluation of the extraction accuracy can by achieved in a straightforward manner. The figure on the left shows the position error - which is the average perpendicular distance of the two corresponding corners of the synthetic squares to the estimated line segment. The figure on the right shows the orientation error - which is the absolute enclosed angle between the normalized line vector of the estimated line and the normal vector of the ground truth edge of the square. The main observation is that the position error remains low even under considerable noise and low contrast (maximum is 0.15 pixel for a contrast of 20 and $\sigma = 4.6$) whereas the orientation error is more sensitive to noise (maximum is 7.2 degrees for a contrast of 20 and $\sigma = 4.6$). Another fact worth mentioning is the observation, that by fitting straight 2D lines to sets of collinear points the location and orientation errors can be significantly reduced, compared to the source primitives - this an inherent property of the least squares fitting model.

inferior accuracy of the TCM is caused by the indirect approach that detects the dominant Thales circle using a RANSAC approach, whereas Rother's method performs a direct search on the line intersections. Since both methods are used for vanishing point detection the processing time is more crucial than the location accuracy. Given the fact that

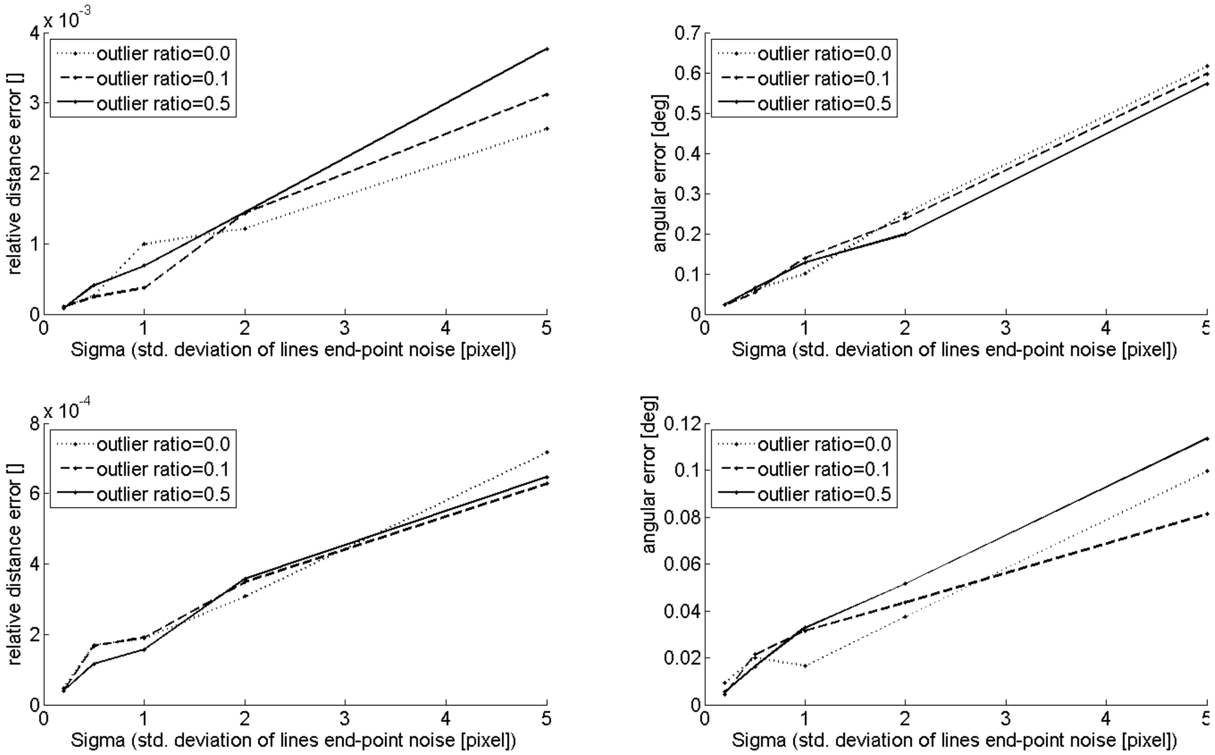
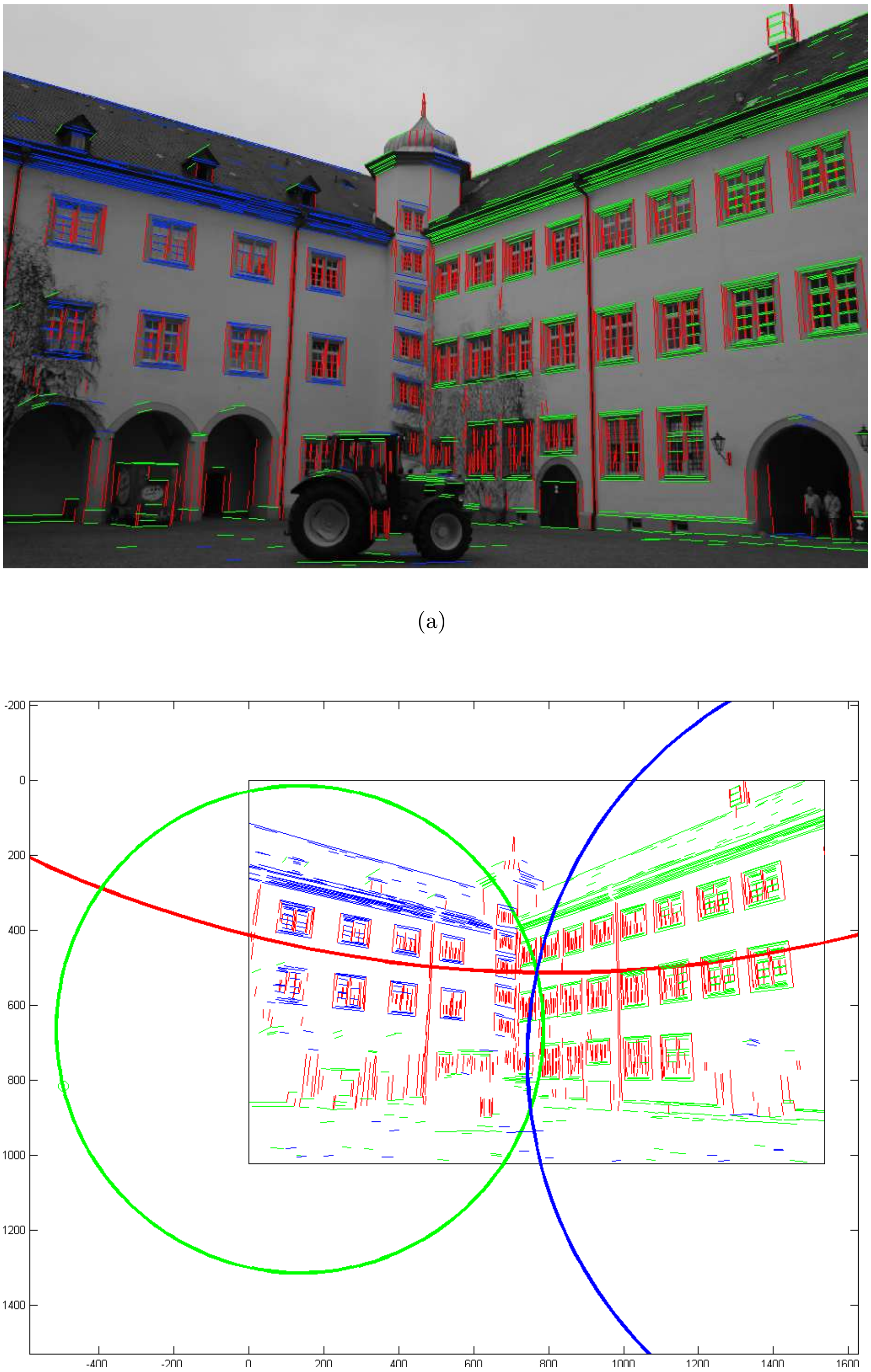


Figure 5.10: Results for the vanishing extraction. In order to simulate realistic imaging conditions the end points of the straight 2d line segments are perturbed by increasing amounts of additive Gaussian noise ($\sigma = 0.1 \ldots 5$) and increasing amounts of random outlier lines are added - the outlier rate or assumes the values $0.0, 0.1$ and 0.5. The figure on the left shows the relative position error for the Thales circle method and the right plot shows the relative position error for Rother's method [103]. The relative location error is the measured distance of the located vanishing point from the images principal point normalized by the true distance. The main observation is that the relative position error for Rother's method is approximately half as high as for the Thales circle method.

for high precision applications a subsequent least squares optimization of the vanishing point position is a standard procedure, the lower accuracy of the Thales circle method is outweighed by it's speed: the Thales circle method is approximately 25 times faster than Rother's method.

The second experiment is carried out on real facade images and here the evaluation is restricted to the visual verification of the detection results. Figures 5.11 an 5.12 shows the extraction result for a typical facade image.

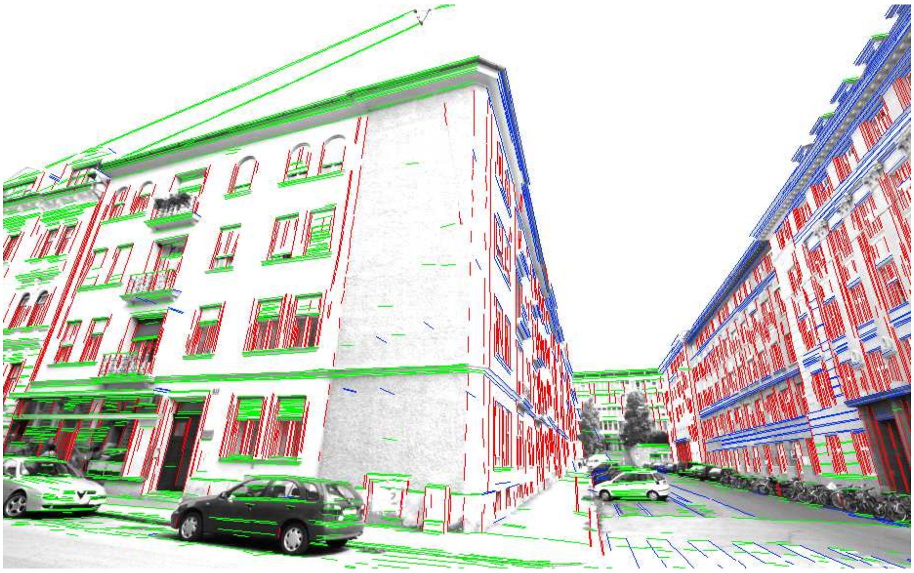
For the experiments with real data the number of RANSAC samples for the Thales circle detection was to five times the number of extracted lines. The reference point was coincident with the image center. With these settings the method was able to robustly identify the dominant vanishing in 40 test images.



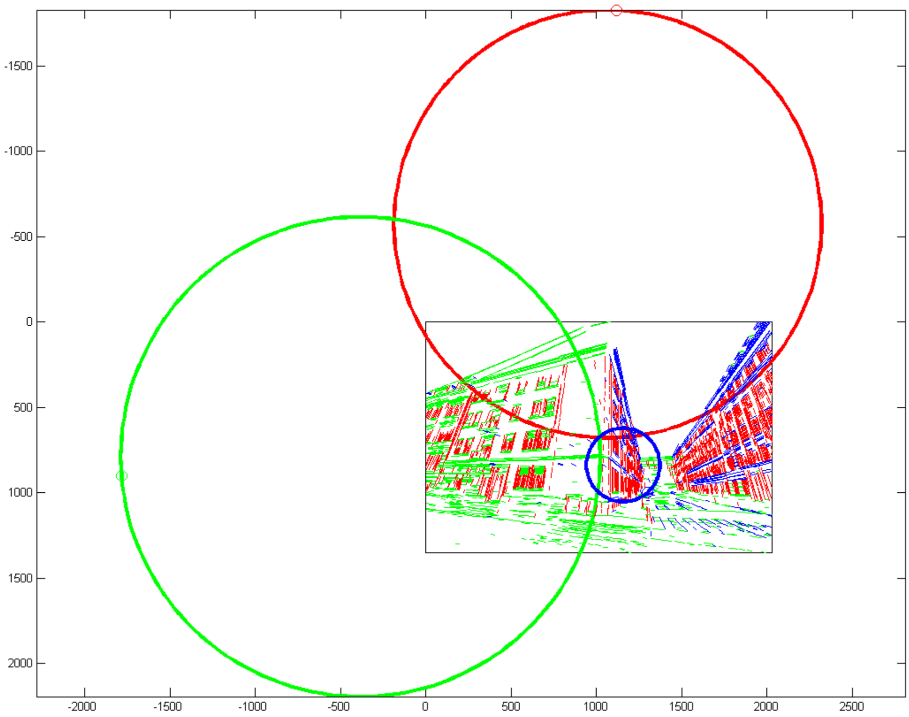
(a)

(b)

Figure 5.11: Example1: Illustration of extracted vanishing points for a real facade image. The top image shows the inlier lines for the three detected vanishing points overlaid on the original image. The bottom image shows the inlier lines and their corresponding Thales circles, the vertical vanishing point (illustrated by the red line segments) and the right horizontal vanishing point (shown in blue) lie far outside the image border, but the left horizontal vanishing point (shown in green) lies close to the image so its Thales circle is fully visible.



(a)



(b)

Figure 5.12: Example2: Illustration of extracted vanishing points for a real facade image. The top image shows the inlier lines for the three detected vanishing points overlaid on the original image. The bottom image shows the inlier lines and their corresponding Thales circles, the vertical vanishing point (illustrated by the red line segments) and the right horizontals vanishing points (shown in blue and green). Since all vanishing points lie relatively close to the image border, the corresponding Thales circles are fully visible.

5.4 Accuracy of affine square fitting

For testing the performance of the affine square fitting we start with an evaluation of the method on synthetic points sets as shown in the introduction of the framework 5.1. The square fitting method faces the challenge of determining the affine transform parameters from a noisy point set that contains outliers. The fitting process is a two stage approach: In the first step a robust ellipse is detected for the point set. The inlier points for this ellipse are then used to estimate an initial solution and in the second stage all points are used to refine the affine square parameters. This approach helps to deal with gross outliers and provides a robust initialization.

The estimated parameters are then compared with the known ground truth. For the experiment we generated affine distorted squares and perturbed the point coordinate with Gaussian noise. The standard deviation of the noise ranges from $\sigma = 0.0..0.27$. Additionally parts of the contour are replaced by outlier points - this models the influence of imperfect segmentation. The completeness assumes values of $c = 1.0, 0.85, 0.7$ resulting in outlier rates of $0\%, 15\%$ and 30%. Figure 5.13 shows the mean errors for the translational, the scale parameters, the rotation angle and the shear factor.

The translational and scale errors are the average Euclidean distance between the true position t_x, t_y and scale s_x, s_y and the estimated position and scales t'_x, t'_y, s'_x, s'_y. So the scale error is computed as the Euclidean distance $e_s = \sqrt{(s_x - s'_x)^2 + (s_y - s'_y)^2}$, the translational error is computed in the same way. The rotation error and the shear error are computed as absolute difference between the ground truth value and the estimated value. For low outlier rates the estimated parameters are close to the ground truth values, but for an outlier rate of 30% the errors increase significantly. This is caused by the fact that one complete side of the square is replaced by outlier points. Thus the estimation of the parameters is strongly influenced by the outlier points. However if the outlier points are more evenly distributed along the synthetic contour, the parameter estimation is likely to be more robust. We chose the present outlier simulation to mimic the typical errors that are present in contours that are generated by segmentation methods.

Figures 5.14 and 5.15 shows fitting results for the synthetic data.

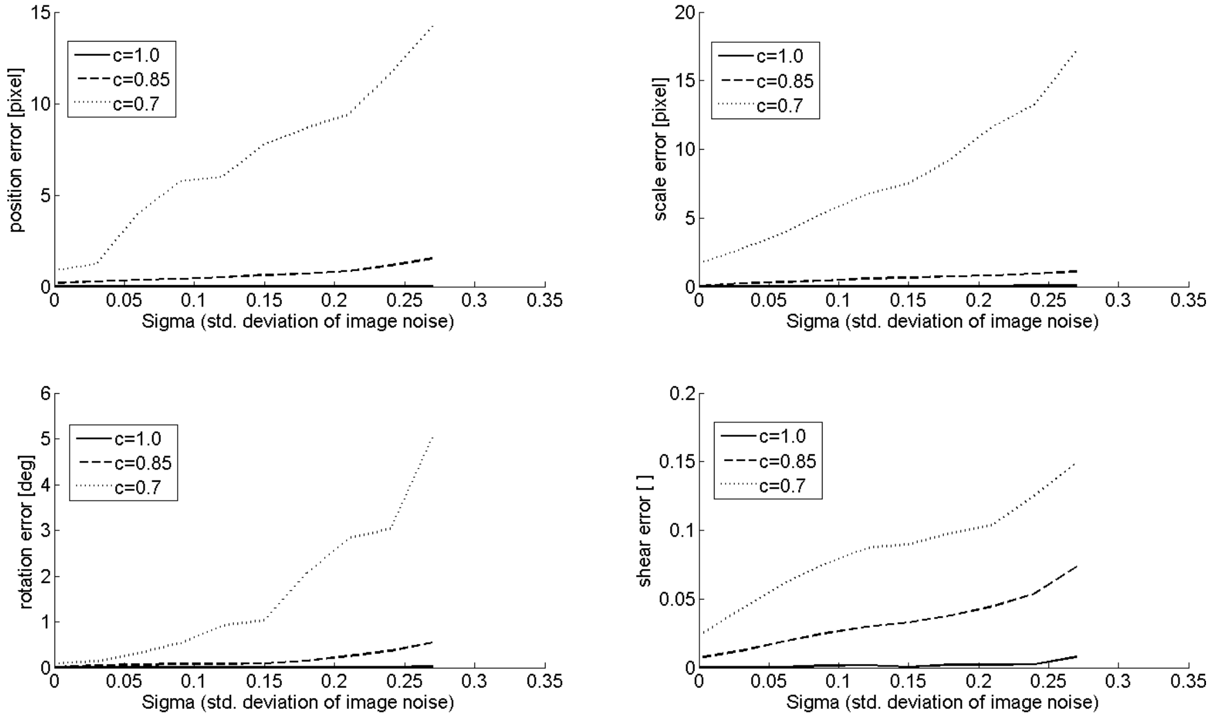
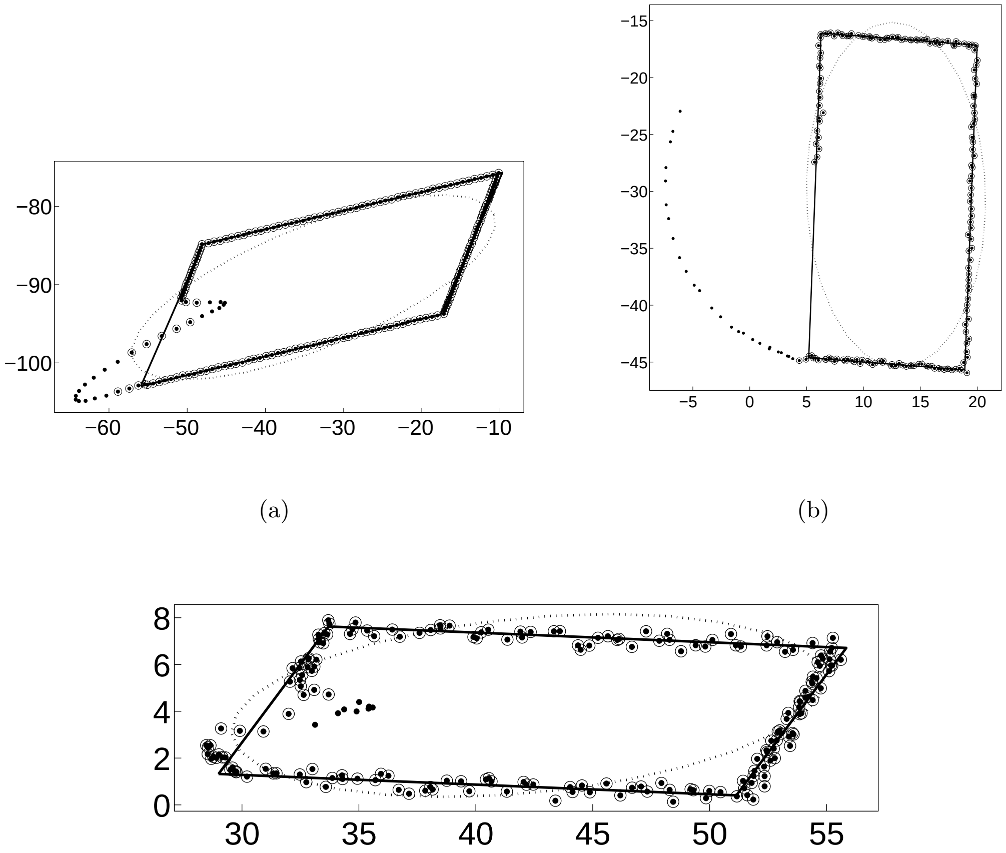


Figure 5.13: Results for the affine square fitting. The four plots show the errors of the translation, scale rotation and shear. The translational and scale errors are the average Euclidean distance between the true position t_x, t_y and scale s_x, s_y and the estimated position and scales t'_x, t'_y, s'_x, s'_y. The rotation error and the shear error are computed as absolute difference between ground truth value and estimated value. The standard deviation of the noise ranges from $\sigma = 0.0 \ldots 0.27$. The completeness assumes values of $c = 1.0, 0.85, 0.7$ resulting in outlier rates of $0\%, 15\%$ and 30%.

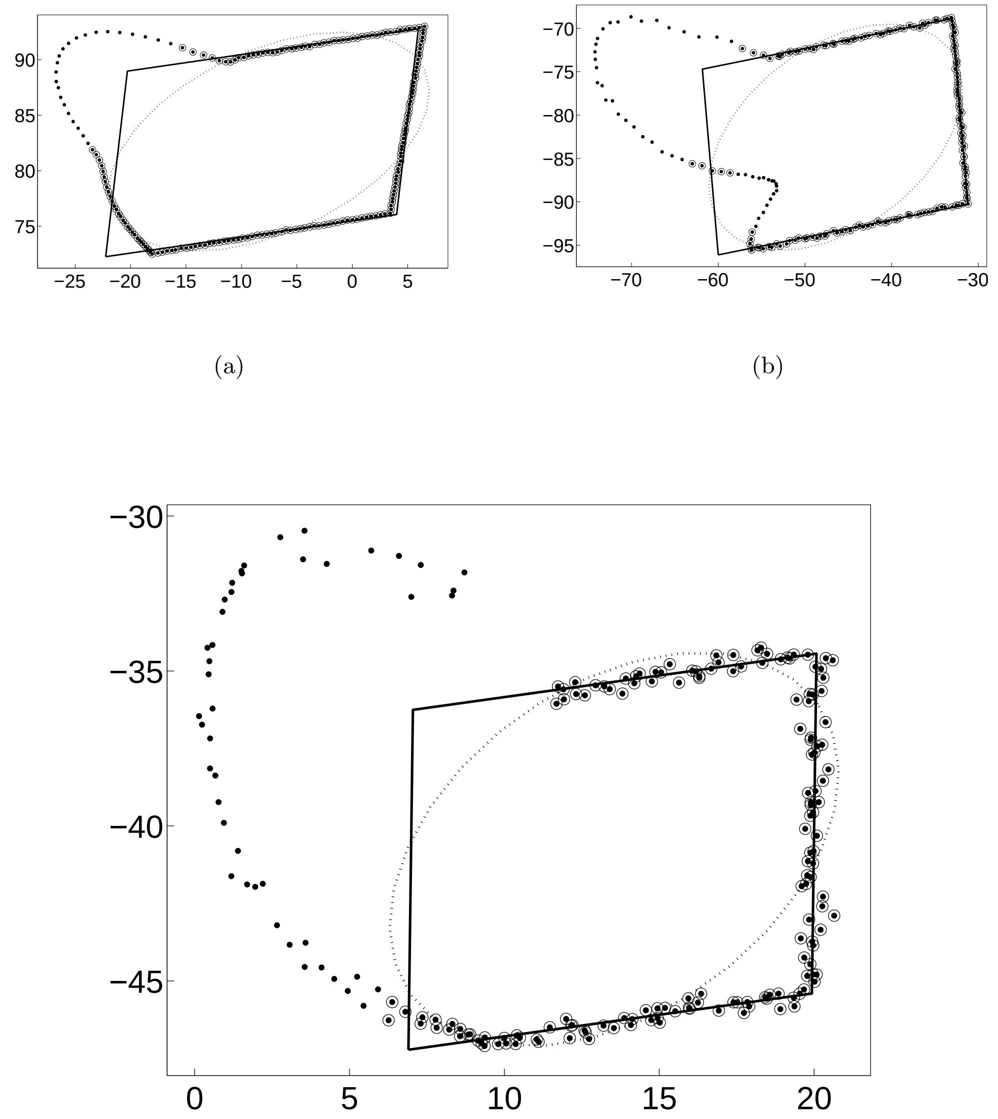


(a)

(b)

(c)

Figure 5.14: Illustration of the affine square fitting approach on synthetic 2D point sets. The number of outlier points is 15% and the point locations are perturbed by Gaussian noise (left $\sigma = 0.03$, right $\sigma = 0.12$, lower $\sigma = 0.24$). The dots represent the initial point set and the circled points are the detected inlier points. The detected ellipse is shown as dotted line and the fitted affine square is drawn using continuous lines.



(a)

(b)

(c)

Figure 5.15: Illustration of the affine square fitting approach on synthetic 2D point sets. The number of outlier points is 30% and the point locations are perturbed by Gaussian noise (left $\sigma = 0.03$, right $\sigma = 0.12$, lower $\sigma = 0.24$). The dots represent the initial point set and the circled points are the detected inlier points. The detected ellipse is shown as dotted line and the fitted affine square is drawn using continuous lines.

For the experiment with real data we extracted MSER'regions (Maximally Stable Extremal Regions) proposed by Matas et al. [61] from typical facade images and fitted affine squares to the outer contour of this regions. A successful fit is reported if 70% of the squares circumference is covered by inlier points (an inlier point must have a perpendicular distance lower than 10% of the squares scale). The images have a resolution of approximately 3000 × 2000 pixel. The overall processing time is in the range of $7 \ldots 12$ seconds on a 2.1Ghz machine.

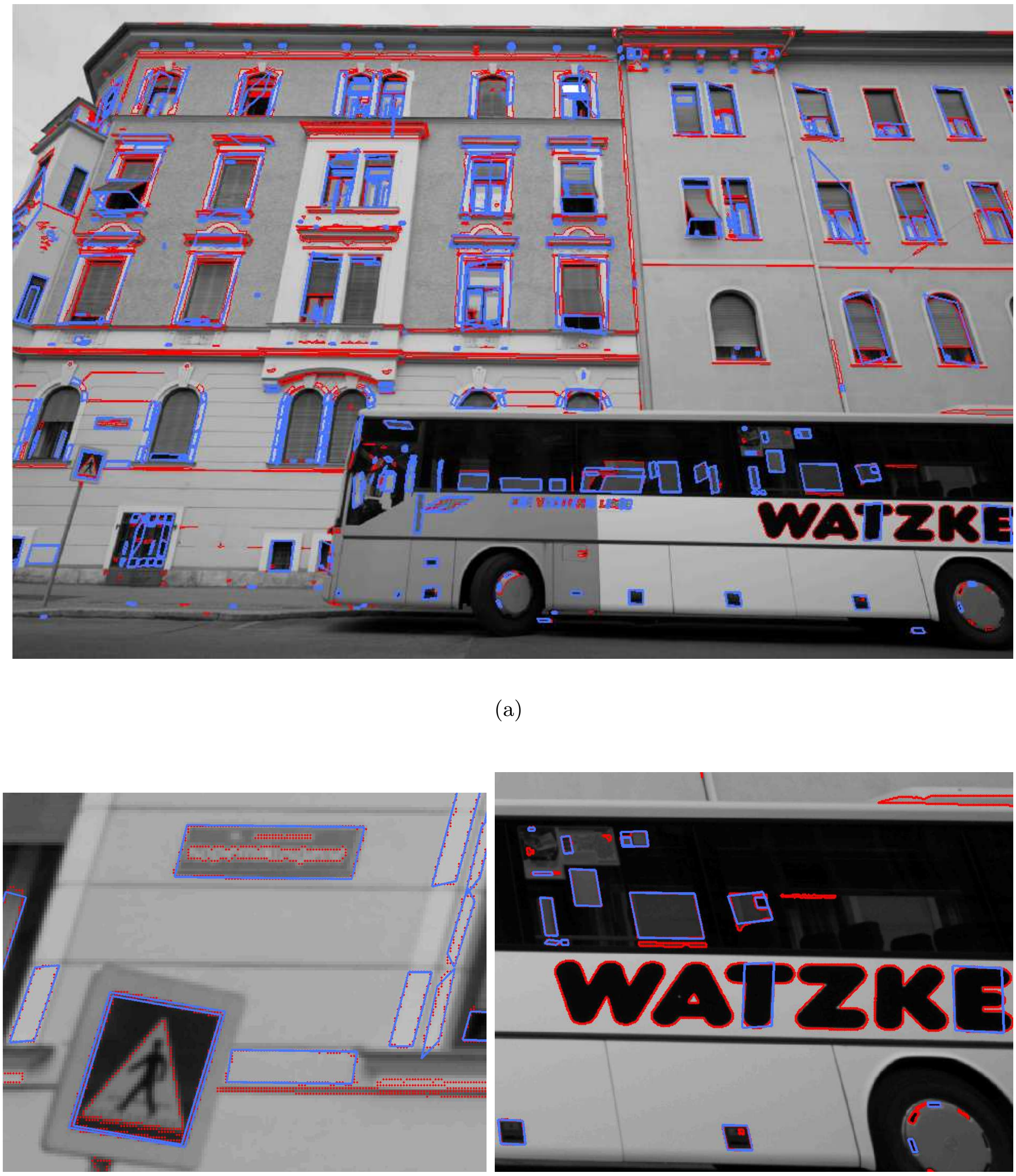
Figures 5.16 and 5.17 show the results of this fitting process. The contours points of the detected MSER's are shown as red dots and the fitted affine squares are shown in blue.

The first image is a typical facade image taken in the city of Graz. The face is highly structured and the detected MSER's often have a complex shape. Thus the detection of the correct transform parameters often fails. However for many contours the visually correct parameters are found, the detail views 5.16(b),(c) show some examples. Note the fitting results for the letters on the side of the bus.

The second image is from a publically available dataset that can be found at *http://cvlab.epfl.ch/ strecha/multiview/denseMVS.html.* In this case the facade is dominated by large windows that contain many small window panes. The fitting method was able to detect many of these rectangular structures.

In conclusion the proposed approach is suitable for detecting rectangular structures in facade images. The affine squares can be used to drive subsequent segmentation and classification methods. Since the fitting approach works on single images only the true aspect ratio of the fitted primitives can not be estimated from the affine transform parameters. This would require further knowledge of the scene, for example perpendicular vanishing points or a 3D facade plane. Especially the presence of a 3D plane would allow to perform a verification step that weeds out bad hypothesis by projecting the affine distorted squares onto the plane and testing for right angles or symmetry.

During the experiments with real data a strong dependency on a robust initialization was observed - a couple of outlier points can have a dominant influence on the estimated parameters. A solution to this problem might be the use of the orientation information that is implicitly present in the contour data. In that case not only the distance of a point to the side of the square but also the angular difference between the points tangent vector and the direction vector of the corresponding side of the square would be taken into account in the fitting process.

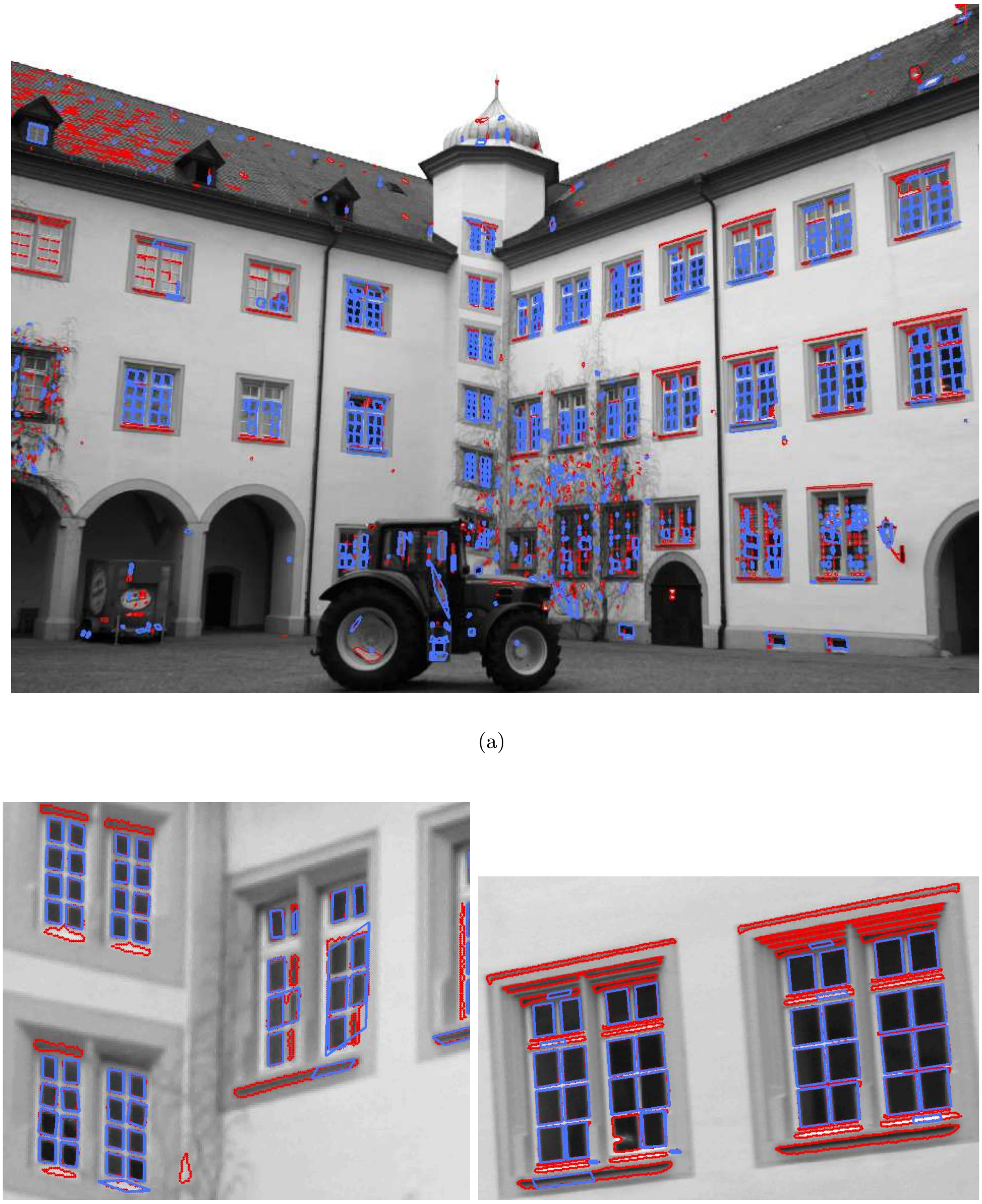


(a)

(b)

(c)

Figure 5.16: Illustration of the affine square fitting approach on a real facade image. The outer contour points of the MSER regions are shown as red dots and the fitted affine square is overlaid in blue. Note that many rectangular details such as traffic signs or details on the bus are detected correctly.



(a)

(b)

(c)

Figure 5.17: Illustration of the affine square fitting approach on a real facade image. The outer contour points of the MSER regions are shown as red dots and the fitted affine square is overlaid in cyan. Note that many of the window panes are detected correctly. For objects that are small with respect to the image dimensions, the affine transform is a good approximation for the perspective distortion effects. Due to the fact that the fitting approach works on single images only and the true aspect ratio of the fitted primitives can not be estimated. This would require further knowledge of the scene e.g. perpendicular vanishing points or a 3D facade plane.

5.5 Evaluation of combined descriptors

In this section the combination of local and global descriptors is evaluated.

The evaluation of the descriptors is performed on several image pairs of different buildings. For each building pair we perform the matching step using all descriptors. All descriptors are computed for Harris corners that are extracted from each image in a preprocessing step. As a reference descriptor we use a 15 × 15 pixel image patch that is sampled w.r.t. a key orientation to achieve rotation invariance. The orientation for each corner point is computed as described in [78]. We perform the matching between the image pairs using normalized cross correlation as a similarity function for descriptor comparison. In order to constitute a valid correspondence, the correlation between two descriptors must be larger than 0.9 and the back matching must also be successful (i.e. the matching with the best matching descriptor, detected in the right image by forward matching a query descriptor from the left image to all descriptors in the right image, must again detect the query descriptor as best match in the left image). For those test images that contain a dominant planar structure we use a RANSAC method to detect the best fitting homography that relates the correspondences. The distance threshold for inlier points was set to 15 pixel - this has the effect that image correspondences with small depth variations on a generally planar object have less influence on the robust estimation of the underlying homography. For image pairs that depict a non planar structure we compute the fundamental matrix with a robust RANSAC scheme. The number of inliers that fulfill the geometric constraint (either homography or epipolar constraint) and the ratio of outlier points to inlier points is used to measure the performance of the particular descriptor.

The first test is carried out on a set of images depicting a planar wall showing graffiti paintings. This is one of the data sets used in the evaluation framework of local descriptors presented in [85]. The images are taken from varying viewing angles and slight rotation of the camera.

Since the wall is approximately planar we computed the best fitting homography from the image correspondences in order to determine the inlier correspondences. Tables 5.1 and 5.2 show the number of total matches, the number of inliers and the ratio of inlier vs. total matches.

When matching the descriptors of image a with the descriptors of image b the reference method (cross correlation) performs best. This is due to the small view point change. When matching image a and image c the pseudo global method performs best. The shape context approach detects the largest number of total matches in both cases, but the

Method	total matches	inliers	ratio
cross correlation	465	428	0.92
shape context	1537	548	0.35
pseudo global method	911	703	0.77

Table 5.1: Results for matching descriptors of image (a) with descriptors of image (b).

Method	total matches	inliers	ratio
cross correlation	174	119	0.68
shape context	1117	144	0.12
pseudo global method	456	415	0.91

Table 5.2: Results for matching descriptors of image (a) with image (c)

number of inliers is always worst.

The second test is carried out on three images of the historical facade of the national library at Josefsplatz in Vienna. The facade has two dominant planes but the planes exhibit strong depth deviations, so a robust fundamental matrix estimation is used to determine the inlier correspondences.

Method	total matches	inliers	ratio
cross correlation	603	512	0.84
shape context	2611	550	0.21
pseudo global method	815	630	0.77

Table 5.3: Results for matching descriptors of image (a) with image (b).

Method	total matches	inliers	ratio
cross correlation	315	220	0.69
shape context	1669	153	0.092
pseudo global method	427	280	0.65

Table 5.4: Results for matching descriptors of image (a) with image (c).

The tables 5.3 and 5.4 for this data set show that the reference method is performing well, the cyclic string matching and the shape context have poor performance and the pseudo global approach yields an acceptable inlier rate of 65 to 77 percent.

This data set is one of the data sets used in the evaluation framework of local descriptors presented in [85]. The images are taken from different distances and considerable rotation of the camera. Since the scene is dominated by an approx. planar facade we computed the best fitting homography from the image correspondences in order to determine the inlier correspondences.

Method	total matches	inliers	ratio
cross correlation	1658	715	0.43
shape context	2871	397	0.14
pseudo global method	1728	723	0.42

Table 5.5: Results for matching descriptors of image (a) with image (b).

Method	total matches	inliers	ratio
cross correlation	153	87	0.56
shape context	1355	failed	failed
pseudo global method	507	465	0.91

Table 5.6: Results for matching descriptors of image (a) with image (c).

Table 5.5 shows that all methods are robust against image rotation and while cross correlation has the best inlier vs. total matches ratio, the pseudo global method yields the largest number of inliers. Table 5.6 demonstrates, that cross correlation is very sensitive against scale changes and shape context failed to detect a valid homography. The pseudo global method performs best and also the cyclic string matching has an inlier vs. total matches ratio of 75 percent.

This data set consists of three images of the Mars statue at the Landhaus in Graz taken from significantly different viewpoints. The scene has a large depth variation and no dominant planar structures are present. For the determination of the inlier points we compute the fundamental matrix.

Method	total matches	inliers	ratio
cross correlation	427	371	0.87
shape context	1586	61	0.03
pseudo global method	654	528	0.81

Table 5.7: Results for matching descriptors of image (a) with image (b)

Method	total matches	inliers	ratio
cross correlation	134	49	0.36
shape context	1256	failed	failed
pseudo global method	224	72	0.32

Table 5.8: Results for matching descriptors of image (a) with image (c)

Table 5.7 shows that the cross correlation performs well under weak view point changes and that the pseudo global method and the cyclic string matching method both yield a high number of inliers and a good inlier vs. total matches rate. In Table 5.8 it becomes obvious that the shape context method is not sufficient to handle large view point changes and that the other methods deteriorate under wide baseline setups.

This data set depicts three images of a modern facade with many repeating patterns. Due to the planarity of the object we use a robustly estimated homography to determine the number of inliers.

Method	total matches	inliers	ratio
cross correlation	269	171	0.63
shape context	2025	126	0.06
pseudo global method	308	223	0.72

Table 5.9: Results for matching descriptors of image (a) with image (b)

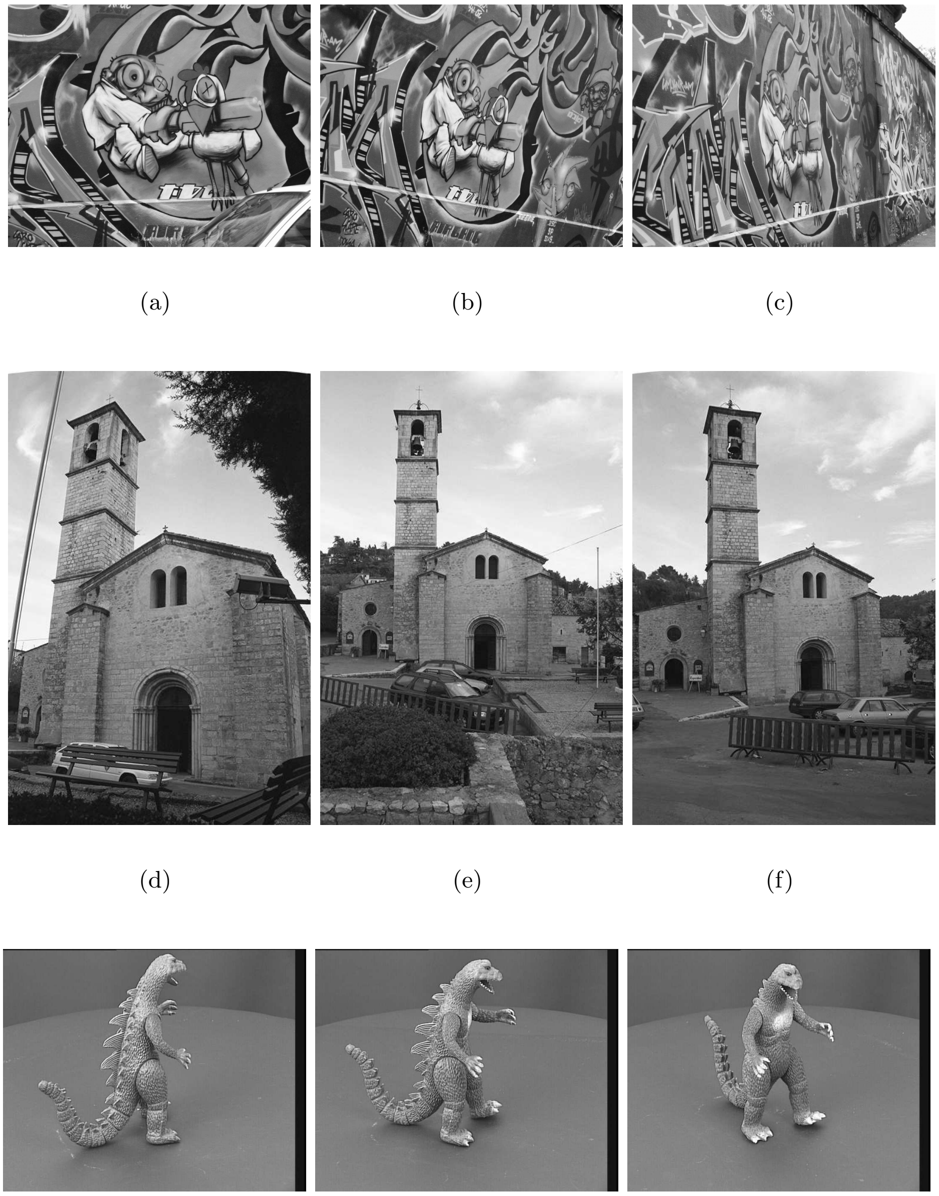
Method	total matches	inliers	ratio
cross correlation	163	39	0.23
shape context	1612	failed	failed
pseudo global method	195	82	0.42

Table 5.10: Results for matching descriptors of image (a) with image (c)

The most noteworthy fact in table 5.9 is that the pseudo global approach performs best and that the cyclic string matching method yields most inlier points but has a low ratio of inliers to total matches. Table 5.10 shows, that the pseudo global approach deteriorates significantly slower than the cross correlation method and the cyclic string matching method failed. The failing can be explained with the large number of nearly identical corner points in the brick wall, that prevents the method from finding unambiguous correspondences.

In this section we presented an evaluation of three methods for the computation of global descriptors. The pseudo global approach performed well for all data sets and yields a good ratio of inliers vs. total matches, which is of significant importance for the robust estimation of the geometric relation between images (epipolar geometry or planar homography). The efficient implementation based on integral images makes it a suitable method for matching large numbers of images, as it is the case in city modeling applications. The cyclic string matching approach yielded the highest number of inliers in five of the ten experiments, but the ratio of inliers vs. total matches is significantly worse when compared with the pseudo global approach. The shape context method performed worst in all experiments, this makes it clear that it is no stand-alone solution for image matching and justifies the combination with a local descriptor as described in [89]. All methods can still be improved and some fine tuning may improve the quality of the results.

In the light of existing methods for descriptor evaluation [85] it is clear that such an evaluation should be carried out in order to have a fair comparison to existing local descriptor approaches. Furthermore our modified version of the cyclic string matching method should be evaluated against the existing one of Tell and Carlsson [123].



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.18: Top row: Three of six images a graffiti scene. Middle row: Three of fifteen images of a church. Bottom row: Three views of the 36 image 'dinosaur' dataset. All data can be downloaded from *http://www.robots.ox.ac.uk/∼vgg/data.html*

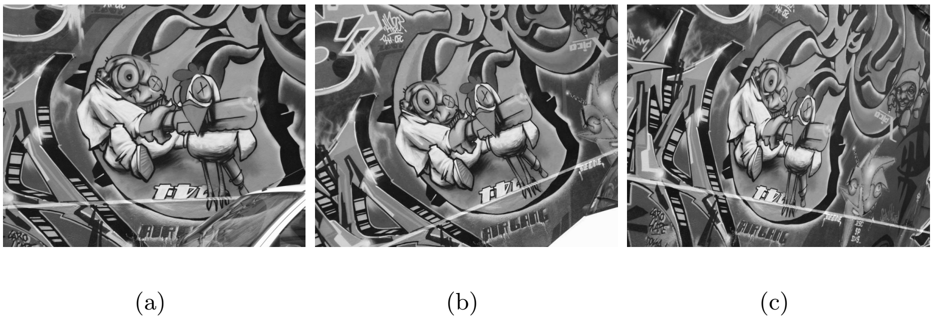


Figure 5.19: Three images of the graffiti scene. *http://www.robots.ox.ac.uk/∼vgg/data.html*

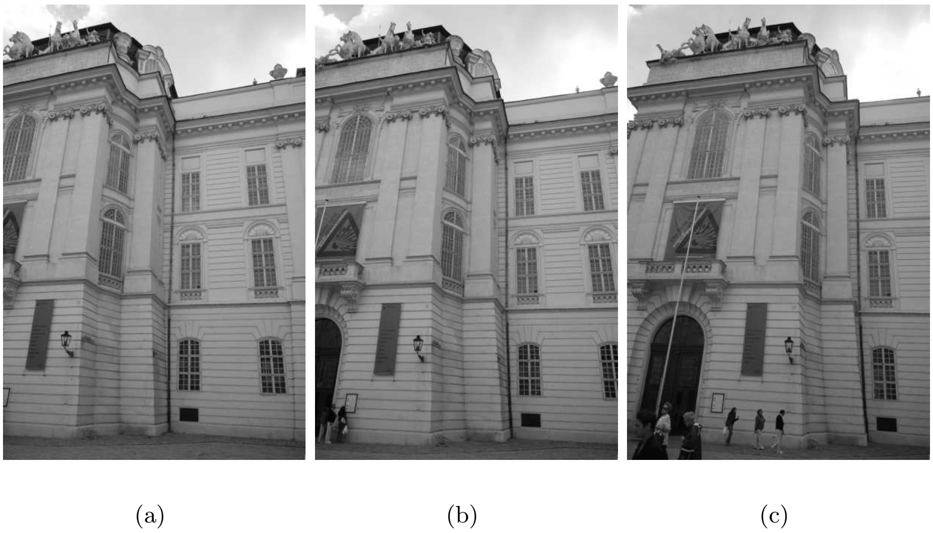


Figure 5.20: Three images of the facade of the national library in Vienna.

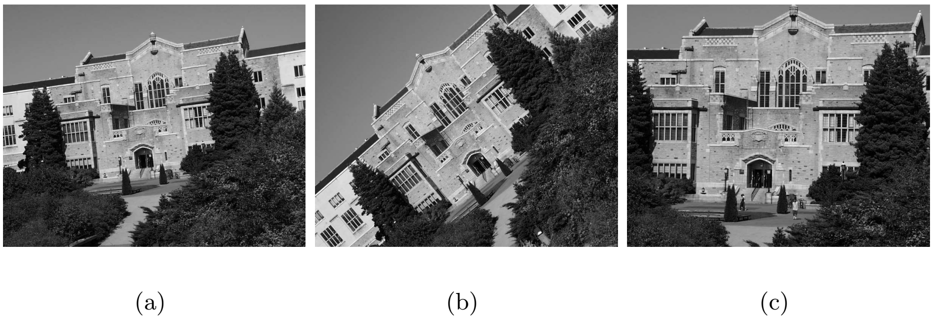
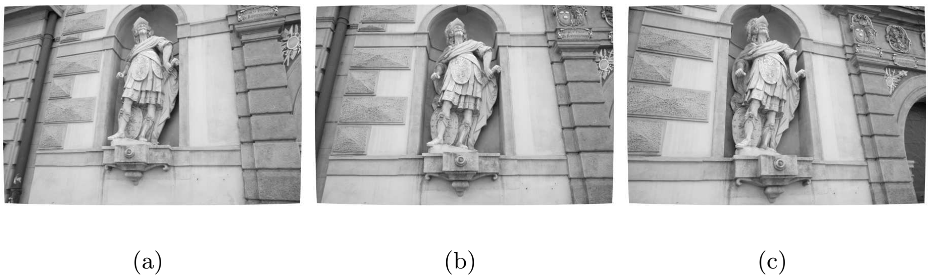
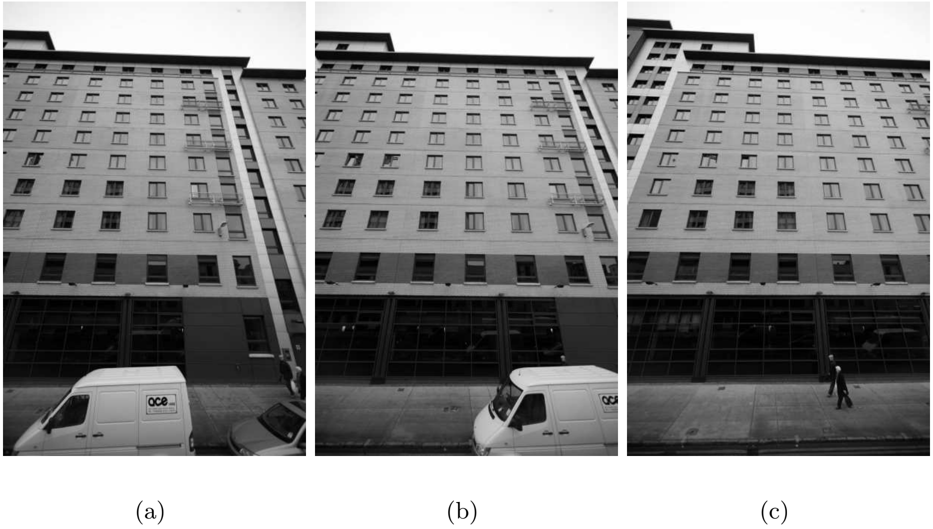


Figure 5.21: Three images of the UBC data set. *http://www.robots.ox.ac.uk/∼vgg/data.html*



(a) (b) (c)

Figure 5.22: Three images of the Mars statue at the Landhaus in Graz.



(a) (b) (c)

Figure 5.23: Three images of a modern facade with many repeating structures.

5.6 Accuracy of 3D modeling

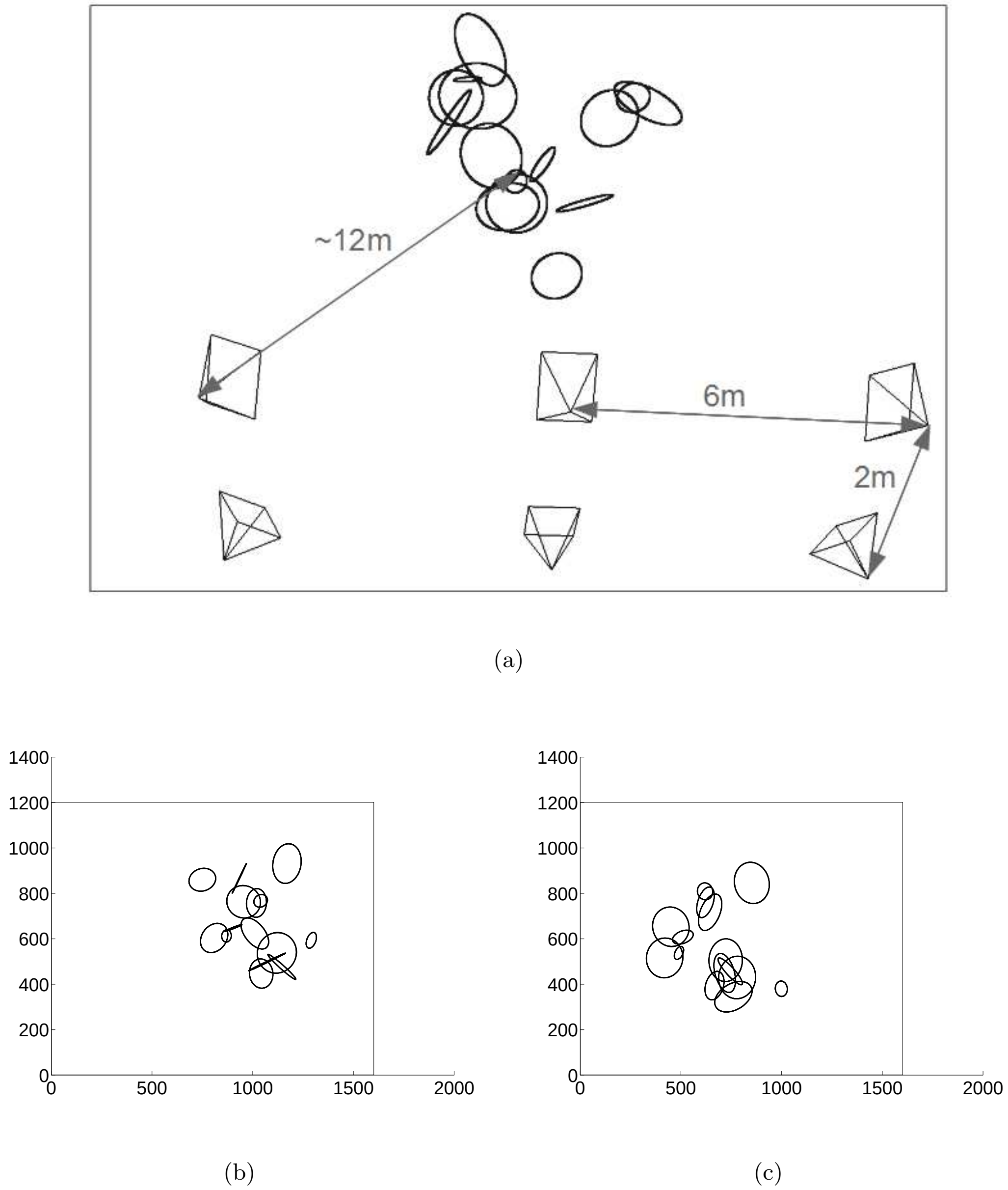
The final set of experiments analyzes the performance of the 3D modeling using the edge sweeping approach from section 4.2. Recall that this method generates 3D hypotheses using a space sweep approach and directed 2D primitives (edgels, ridgels) as input. The evaluation is carried out on synthetic and real data sets. The synthetic data sets are directly generated sets of 2D primitives . In contrast to features that are extracted from synthetically generated images this approach has the advantage that all geometric errors of the features are known.

For the experiments with real image data the error propagation method is used to estimate the position and orientation uncertainty of the reconstructed 3D primitives.

For each dataset the uncertainties are illustrated by histograms, whereas the first two histograms show the 3D uncertainty of the reconstructed primitives and the rightmost histogram shows the distribution of the residuals in image space.

5.6.1 Synthetic data

The first of the following three experiments will investigate the effect of position errors and orientation errors of the extracted primitives. The synthetic setup for this experiment consists of six cameras that view 15 3D circles. The cameras are arranged in two rows of three cameras. The baseline in the row is three meter and the two rows are separated by 2 meter. This arrangement ensures a vertical and horizontal parallax. The 3D circles are located at distances between 9 and 15 meters from the cameras. The focal length of the synthetic cameras is 1200 pixel, the image resolution is 1600 × 1200 pixel and the principal point is at $(800, 600)$. The image edgels are generated by projecting the 3D points of the circles into the image space and adding Gaussian noise to the 2D position and the orientation (the normal vector of the edgel). Figure 5.24 shows a 3D rendering of the setup and two illustrations of the 2D image data.



(a)

(b)

(c)

Figure 5.24: Illustration of the setup for generating the synthetic edgel data. The upper image shows a 3D rendering of the setup: Six cameras (symbolized by their frustums) view a set of fifteen 3D circles. The circles are projected into the cameras to generate the synthetic edgels. The images have a resolution of 1600 × 1200 pixel. Subfigures (b) and (c) show the projected edgels for two different cameras.

Thus the noisy position p' of an edge is its true position p plus a random offset: $p' = p + (\mathrm{d}x, \mathrm{d}y)^T$, where $\mathrm{d}x$ and $\mathrm{d}y$ are uniformly distributed random values from the interval $(-e_{pos}, e_{pos})$ and e_{pos} varies from 0.1 to 0.5 pixel. The noisy normal vector v' is the true normal v perturbed by a random angular offset e_{ori}, where e_{ori} varies from 1 to 10 degrees.

Figures 5.26, 5.27 and 5.28 show the error histograms. The left histogram shows the distribution of the 3D position errors, the histogram in the middle shows the distribution of the 3D orientation errors and the right histogram shows the distribution of the image residuals (== distance from re-projected 3D primitive to directed 2D feature).

For the given synthetic data the position error of a reconstructed primitive is the Euclidean distance to the closest point on the corresponding 3D circle. The general idea is illustrated in figure 5.25. For this projection the reconstructed primitive is first projected onto the 3D plane that is defined by the 3D circle and the projection onto the circle is then achieved by determining the circle's intersection with the chord that spans from the circle's mid-point to the projected point.

The orientation error is determined as the enclosed angle between the circle's tangent (at the projected point) and the direction vector of the reconstructed primitive. For the determination of outlier points we set a threshold of 10mm, that means every 3D primitive that lies further from its corresponding circle is classified as outlier.

The main observation is that the reconstruction accuracy strongly depends on the accuracy of the extracted features. The introduction of a fixed threshold for the classification of outliers is somewhat arbitrary and a dynamic threshold that takes the noise levels into account would allow to make a better distinction between outliers and inliers. For the next two experiments we set the noise level for the position to the maximal value ($e_{pos} = 0.5\,pixel$ and the noise level for the orientation to the minimal value $e_{ori} = 1\ degree$ and vice versa ($e_{pos} = 0.1\ pixel, e_{ori} = 10\ degrees$). This allows us to analyze which error type has a stronger influence on the reconstruction accuracy.

Figures 5.29 and 5.30 again show the error histograms for the position errors and the orientation errors of the reconstructed 3D primitives.

The most striking observation of the last two experiments is that the reconstruction accuracy mainly depends on the position accuracy of the extracted primitives, less so on the orientation accuracy. Even with an orientation noise level of $e_{ori} = 10\ degrees$ the reconstruction accuracy of most of the primitives is below $\pm 3mm$, the mean is $\pm 1.14mm$. In contrast, for the experiment with the position noise level set to its maximal value

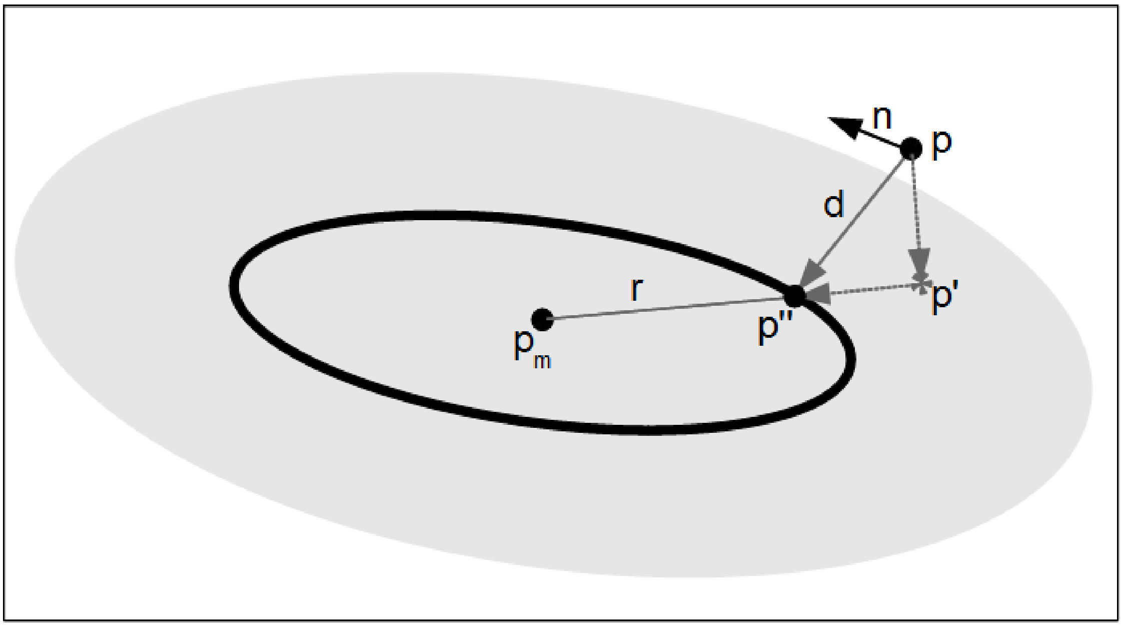


Figure 5.25: Illustration of the 3D position error of a reconstructed point p with respect to a 3D circle: In order to compute the distance d, the point is first projected onto the plane that is defined by the circle (shown in gray) resulting in point p'. The projection onto the circle is then computed as: $p'' = r\frac{p'-p_m}{\|p'-p_m\|}$, where m_p is the mid-point and r is the radius of the circle. The distance d follows as $d = \|p - p''\|$. The orientation error between the direction vector n associated with point p and the circle is the enclosed angles between n and the circles tangent vector at p''.

$e_{pos} = 0.5\ pixel$, the reconstruction accuracy drops to an average value of $\pm 7.12mm$. These findings give a strong hint on where to improve in the feature extraction pipeline, namely on the position accuracy of the features.

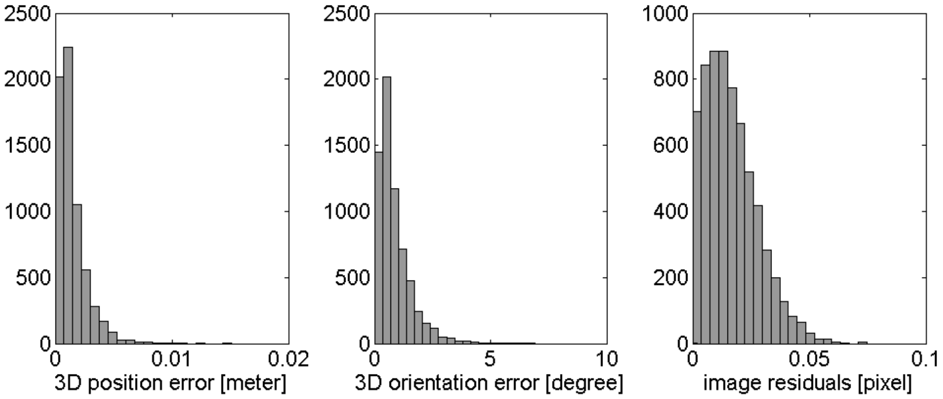


Figure 5.26: Error histograms for the synthetic data set: Fifteen 3D circles are observed by six cameras. The position noise for the edgels is (e_{pos} = 0.1 *pixel*, the orientation noise is e_{ori} = 1 *degree*. From the 7896 edgels in the reference image, 6513 hypotheses are reconstructed, that is a rate of 0.82, 11 of the reconstructed primitives are classified as outliers since they lie more than 10mm from the true position. The left histogram shows the distribution of the 3D position errors, the histogram in the middle shows the distribution of the 3D orientation errors and the right histogram shows the distribution of the image residuals (== distance from re-projected 3D primitive to directed 2D feature).

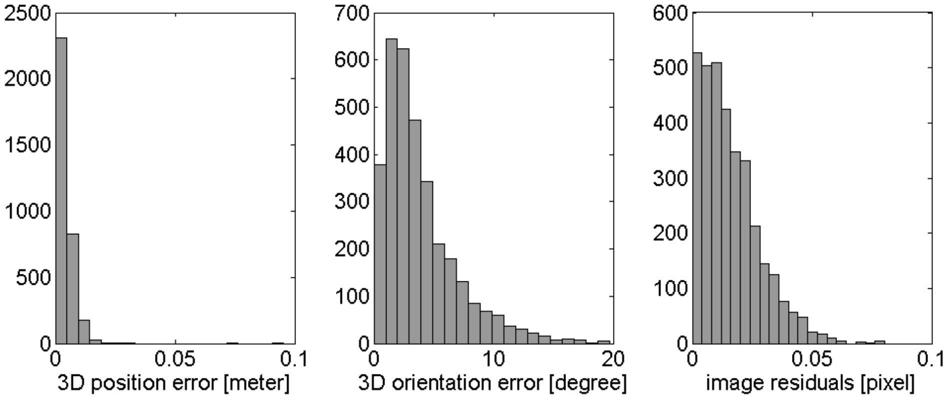


Figure 5.27: Error histograms for the synthetic data set. The position noise for the edgels is (e_{pos} = 0.3 *pixel*, the orientation noise is e_{ori} = 5 *degrees*. From the 7896 edgels in the reference image, 3365 hypotheses are reconstructed, that is a rate of 0.43, 14 of the reconstructed primitives are classified as outliers since they lie more than 10mm from the true position.

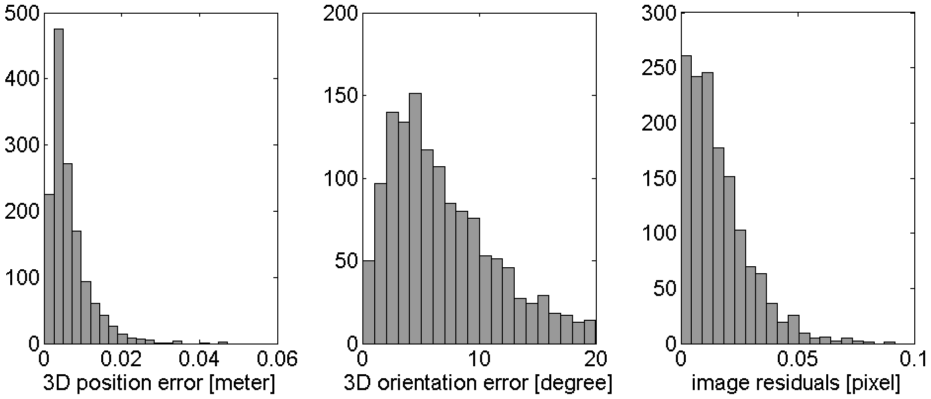


Figure 5.28: Error histograms for the synthetic data set. The position noise for the edgels is ($e_{pos} = 0.5$ *pixel*, the orientation noise is $e_{ori} = 10$ *degrees*. From the 7896 edgels in the reference image, 1425 hypotheses are reconstructed, that is a rate of 0.17, 18 of the reconstructed primitives are classified as outliers since they lie more than 10mm from the true position.

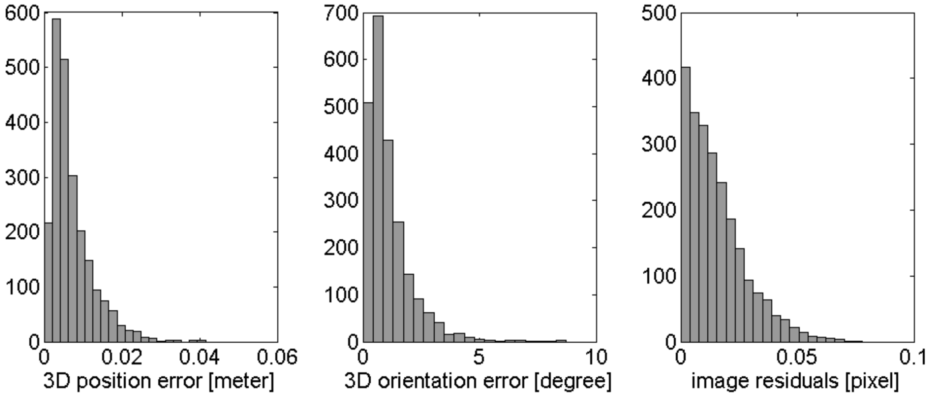


Figure 5.29: Error histograms for the synthetic data set. The position noise for the edgels is maximal ($e_{pos} = 0.5$ *pixel*), the orientation noise is minimal ($e_{ori} = 1$ *degree*). From the 7896 edgels in the reference image, 2313 hypotheses are reconstructed, that is a rate of 0.29, 24 of the reconstructed primitives are classified as outliers since they lie more than 10mm from the true position. The histogram of the position errors and the outlier count indicate that the reconstruction accuracy strongly depends on the position accuracy of the extracted features.

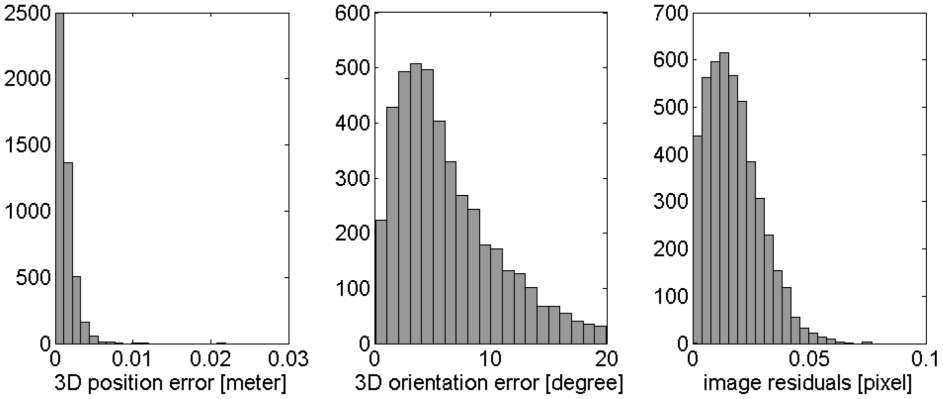


Figure 5.30: Error histograms for the synthetic data set. The position noise for the edgels is minimal ($e_{pos} = 0.1\ pixel$), the orientation noise is maximal ($e_{ori} = 10\ degrees$). From the 7896 edgels in the reference image, 4633 hypotheses are reconstructed, that is a rate of 0.58, 20 of the reconstructed primitives are classified as outliers since they lie more than 10mm from the true position. This experiment shows that the reconstruction accuracy is dominated by the position accuracy of the extracted features

5.6.2 Real data

5.6.2.1 The fountain data

The first experiment with real data uses a multi image data set of a fountain. This data set is part of an evaluation initiative for multi view stereo methods, organized by the universities of Lausanne and Leuven and the Forschungsinstitut Optronik und Mustererkennung in Ettlingen, Germany. The data set is publically available: *http://cvlab.epfl.ch/∼strecha/multiview/denseMVS.html* . The fountain scene was recorded by taking eleven images and in order to generate a ground truth surface, the scene was also recorded with a 3D laser scanner. The Lidar data are represented as closed surface using a triangle mesh. This mesh is also part of the publically available data.

Figure 5.31 shows a rendering of the triangle mesh.

The intrinsic camera parameters are known and exterior orientation parameters are also provided. The images are images are free of lens distortion. Figure 5.32 shows four of the eleven images. The resolution of the images is 3072 × 2048 and the Lidar data consist of 12.99 million triangles. The high density of Lidar points makes the data set well suited for evaluating the proposed feature-based modeling method.

For our experiments we use the first six images of the sequence. The average distance between the camera origins (baseline) is 1.37 meter. The positions of the reconstructed primitives are then compared to the Lidar data, which serve as ground truth. The error of the 3D point with respect to a triangle mesh is either the distance to the closest triangle facet (if the 3D point projects into the facet), or the perpendicular distance to a triangle's edge. Any point that is further than 10mm from the triangle mesh is considered as outlier.

To recapitulate the main steps of the workflow are:

1. Feature extraction; Extract edgel chains from the input images (parameters for Canny edge detection:$\sigma = 1.0$, $t_{low} = 4.0$, $t_{high} = 6.0$, parameters for edge-to-chain linking: minimal chain points = 15).

2. Space sweeping; Extract 3D primitives using the proposed space-sweeping approach (parameters: minimal number of image measurements=4).

3. Evaluation of the 3D position errors by comparing the position of the reconstructed primitives with the Lidar data.

The three plots in figure 5.33 show the error histograms for the position errors. The histograms are computed from the reconstruction errors of three individual runs. In each

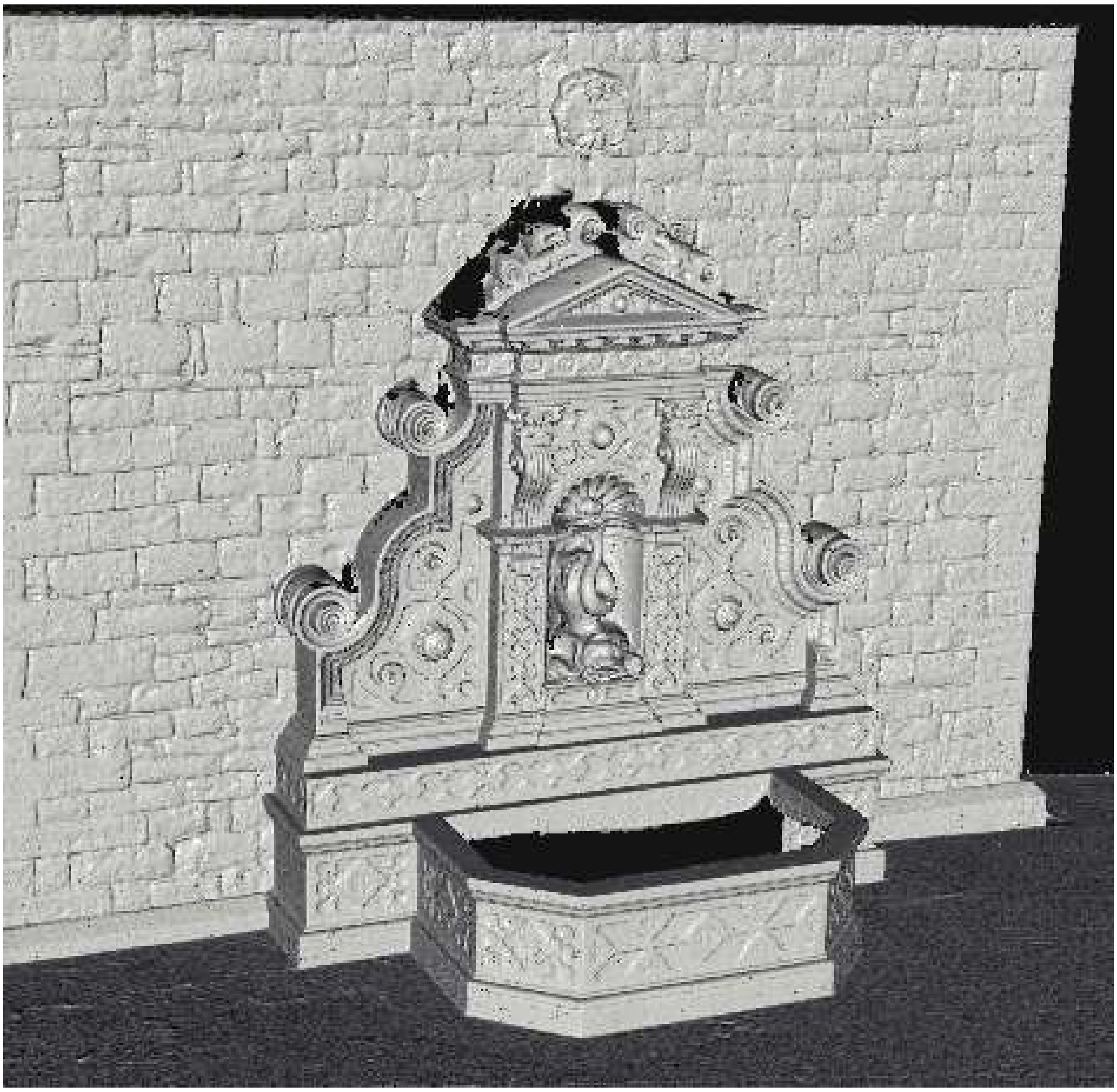
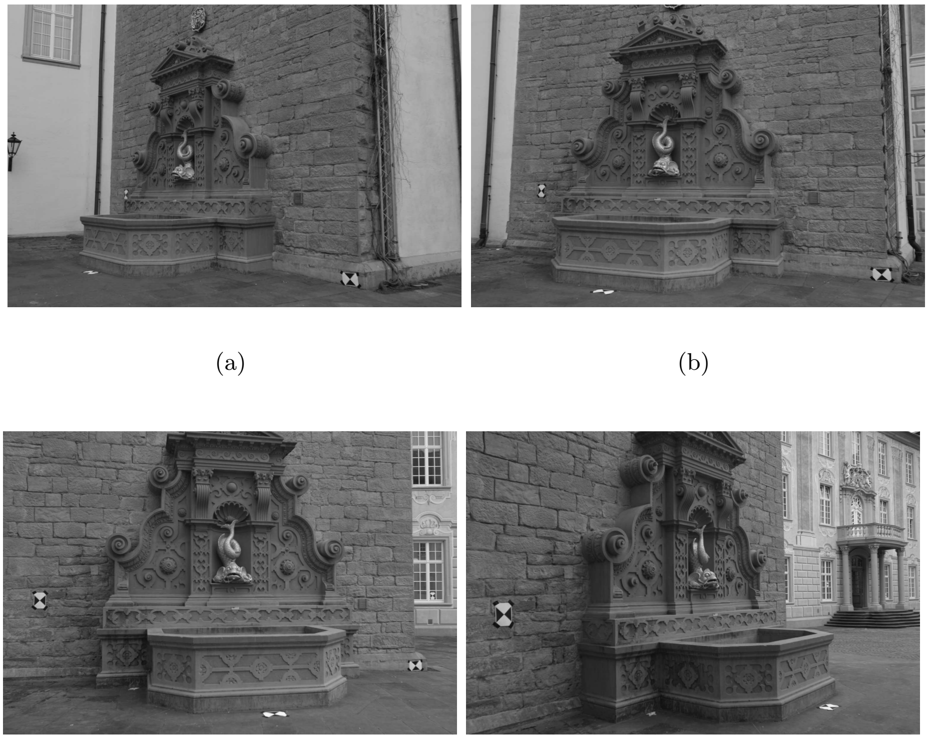


Figure 5.31: Rendering of the triangle mesh of the fountain scene, based on Lidar data. The mesh consists of 12.99 million triangles. The high point density makes this data set well suited for evaluating reconstruction results of image-based 3D modeling methods. The image was taken from *http://cvlab.epfl.ch/∼strecha/multiview/denseMVS.html*

.

run a different image serves as reference image, namely image 2, image 3 and image 4 of the sequence. For reference image 2, 95973 3D primitives are generated (356 points are classifies outliers), for reference image 3, the number of primitives is 103095 (487 outliers), and for reference image 4, 131716 primitives are found (434 outliers). The error distributions look similar and the mean reconstruction errors are ±7.4mm for reference image 2,



(a)

(b)

(c)

(d)

Figure 5.32: Four of the eleven images of the fountain sequence - a publically available multi view data set for evaluation of image-based 3D reconstruction methods (*http://cvlab.epfl.ch/∼strecha/multiview/denseMVS.html*). The images do not depict a typical urban scene but the combination of high resolution images (3072×)2048 and high resolution Lidar data (12.99 million triangles) provide a good basis for evaluating the proposed feature-based modeling method.

±7.5mm for reference image 3, and ±8.7mm for reference image 4. The number of outliers is in approximately the same range for all reference images. Despite the relatively low outlier count compared to the total number of primitives, their presence might influence subsequent processing steps. However the detection of outliers based on their estimated 3D position and orientation uncertainty fails for certain degenerate cases.

The evaluation gives a good estimate of the achievable reconstruction accuracy. It also shows the discrepancy of point densities between the modeling methods: The proposed space sweeping approach produces less than 1% of the data points of the Lidar scanner. This comparison holds also for dense multi-view stereo methods, where the number of generated 3D points is approximately the number of pixels (6.3 MPixels in the present case). However, given the fact that approximately 500.000 edgels are detected in the input images the reconstruction rate is low.

However the experiment is not ultimately conclusive, since no accuracy for the Lidar data is given and the accuracy of the registration of the Lidar coordinate system to the coordinate system of the cameras is also unknown. Another factor that makes the test somewhat inconclusive is the fact that the space-sweeping method is based on edgels and many edgels are detected on sharp depth-discontinuities where the Lidar scanning technology can not perform at its highest accuracy (due to a non negligible spot size of the laser beam).

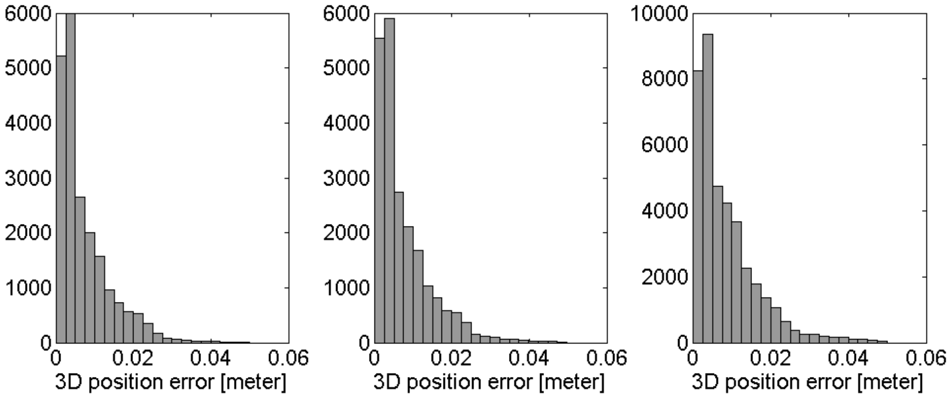


Figure 5.33: Error histograms for the position errors of three individual runs. The left histogram shows the error distribution for image 2 serving as reference image, the histogram in the middle shows the distribution for reference image 3 and the right histogram depicts the error distribution for image 4 serving as reference image. The errors are measured as the perpendicular distance of the reconstructed 3D primitive with respect to the closest triangle facet of the mesh that is defined by the Lidar data points. The error distributions look similar and the mean reconstruction errors are $\pm 7.4mm$ for (a), $\pm 7.8mm$ for (b) and $\pm 9.3mm$ for (c). The evaluation gives a good estimate of the achievable reconstruction accuracy. However the experiment is not ultimately conclusive, since no accuracy for the Lidar data is given and the accuracy of the registration of the Lidar coordinate system to the coordinate system of the cameras is also unknown.

The following experiments use real images that were acquired with digital SLR cameras in the city of Graz. The intrinsic camera parameters are determined by an off-line camera calibration and the exterior camera orientation parameters of the images are determined by automatic multi-image matching followed by an estimation of the relative orientation of image pair in the sequence and a final bundle adjustment for all images. The depth range for the sweeping is automatically calculated from the reconstructed tie-points of the image orientation stage. Due to the fact that for these data sets no ground truth is available, the position and orientation uncertainties of the generated 3D primitives is estimated by error propagation. Thus the workflow is structured as follows: To recapitulate the main steps of the workflow are:

1. Feature extraction; Extract edgel chains from the input images (parameters for Canny edge detection:$\sigma = 1.0$, $t_{low} = 4.0$, $t_{high} = 6.0$, parameters for edge-to-chain linking: minimal chain points = 15).
2. Space sweeping; Extract 3D primitives using the proposed space-sweeping approach (parameters: minimal number of image measurements=4).
3. Estimation of the 3D position and orientation uncertainties by error propagation.

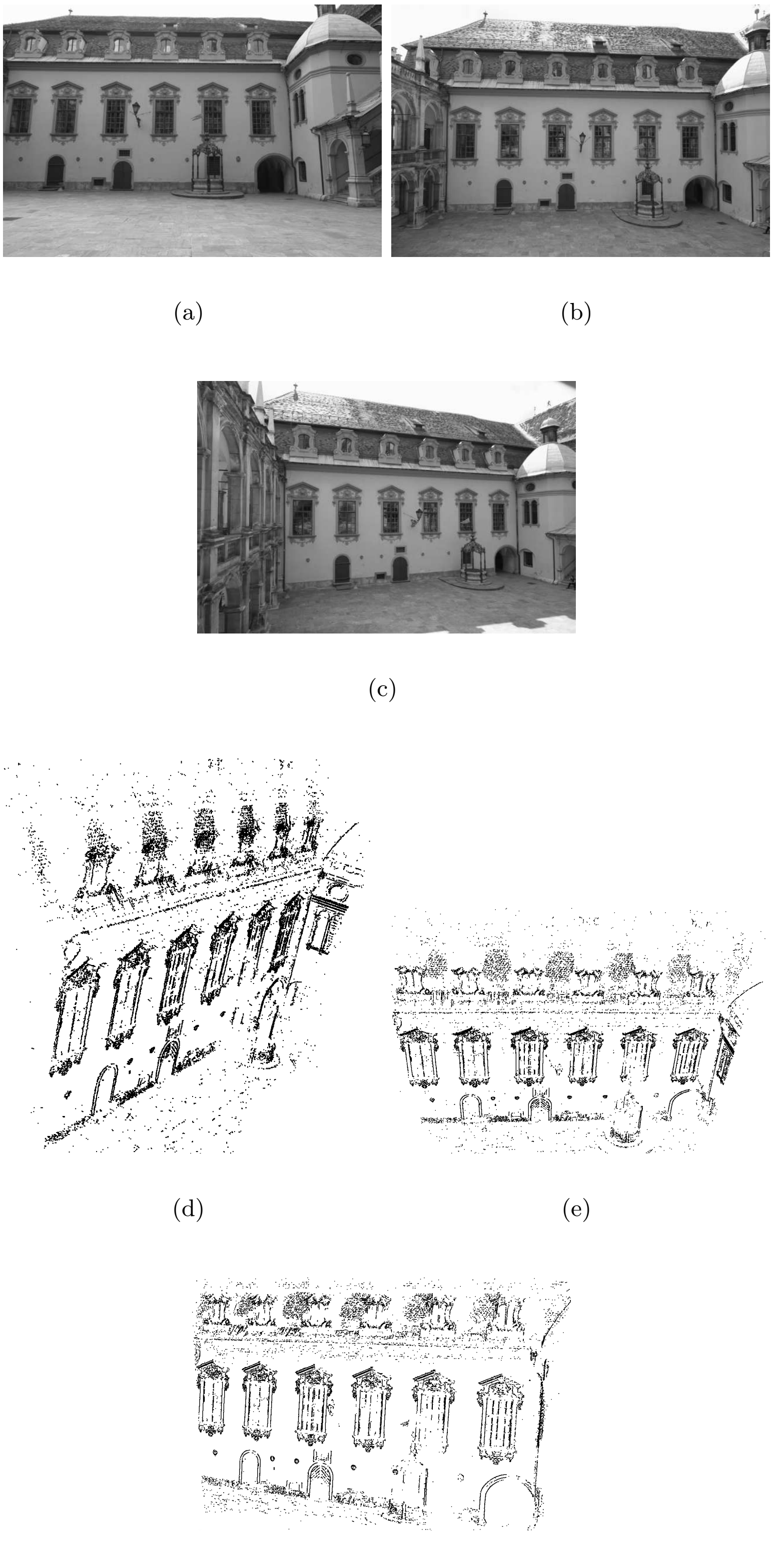
5.6.2.2 The historical courtyard in Graz

The first data set that is investigated consists of five images of the historical Landhaus courtyard in Graz. The images have a resolution of 2160×1440 pixel. The most prominent objects are the historical facade and roof structure and the ironwork that covers the well. Figure 5.34 shows three of the five images and renderings of the reconstructed 3D primitives from different viewpoints. The overall structure of the scene is captured and many details show up in the reconstruction. Especially notable is the partial reconstruction of the ironwork of the well - it was modeled despite its sparse structure.

For the visual assessment of the reconstruction accuracy and completeness the extracted 2D edgels and the projected 3D primitives are overlaid onto the original images. For greater clarity only a small portion of the image is visualized. Four such overlays are shown in figure 5.35. These illustrations serve two purposes: First it gives an impression of the reconstruction accuracy by comparing the extracted 2D edgels position to the reprojected primitives position. Since the criterion for accepting a 3D hypothesis is a low reprojection error the projections of the primitives are always within less than 0.5 pixel from the 2D edgel. The second purpose for this illustration is the estimation of the completeness of the reconstruction, or in other words how many 2D edgels do have a 3D primitive associated with them? In the case of the Landhaus facade many horizontal edgels do not have a 3D primitive associated with them. This is caused by the fact that cameras are displaced in horizontal direction, parallel to the facade and thus horizontal structures cannot be reconstructed with a low uncertainty.

In order to estimate the accuracy of the reconstruction the covariances for the position and orientation are computed by error propagation. For the a priori position uncertainty a value of ± 0.25 pixel was assumed and for the orientation uncertainty of the edgels normal vectors a value of ± 3 degrees was assumed. The mean position uncertainty is $\pm 12.2mm$ and the mean orientation uncertainty is ± 3.9 degrees. The histogram that illustrates the orientation uncertainty is cut at 9 degrees - this was the predefined threshold for accepting a 3D hypothesis during the sweeping.

Figure 5.36 shows the histograms for the a posteriori uncertainties of the position and the orientation.



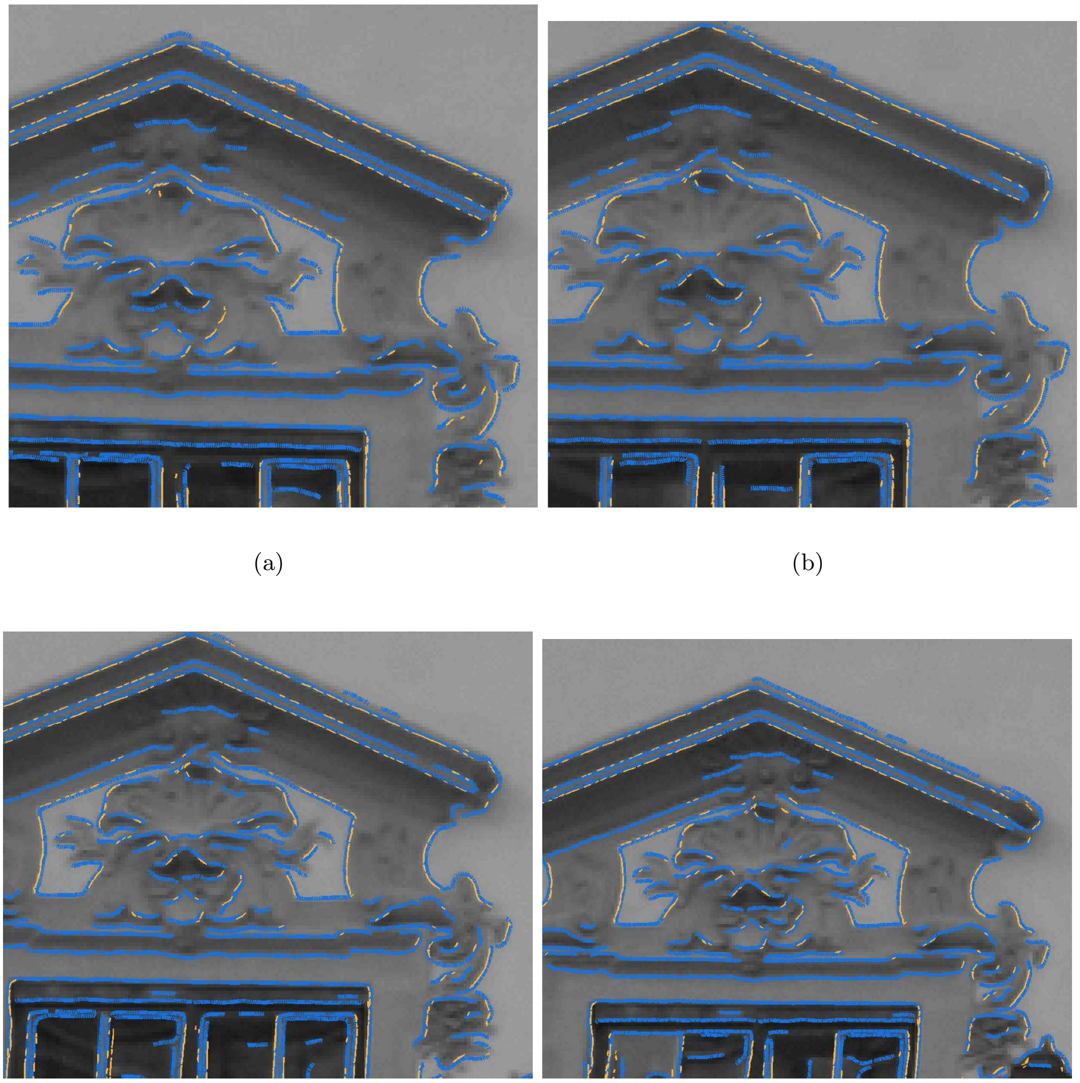
(a) (b)

(c)

(d) (e)

(f)

Figure 5.34: Top row: Three of the five images of the courtyard scene (image size = 2160 × 1440, 210k edgels on average) Bottom row: Three views of the resulting 3D point cloud.



(a)

(b)

(c)

(d)

Figure 5.35: Overlays of extracted 2D edgels (in blue) and projected 3D primitives (in orange) onto four source images. This illustration gives an impression of the reconstruction accuracy by comparing the extracted 2D edgels position to the reprojected primitives position. Furthermore it shows that horizontal features do not have a 3D primitive associated to them - this is due to the geometric configuration of the cameras (only horizontal displacements).

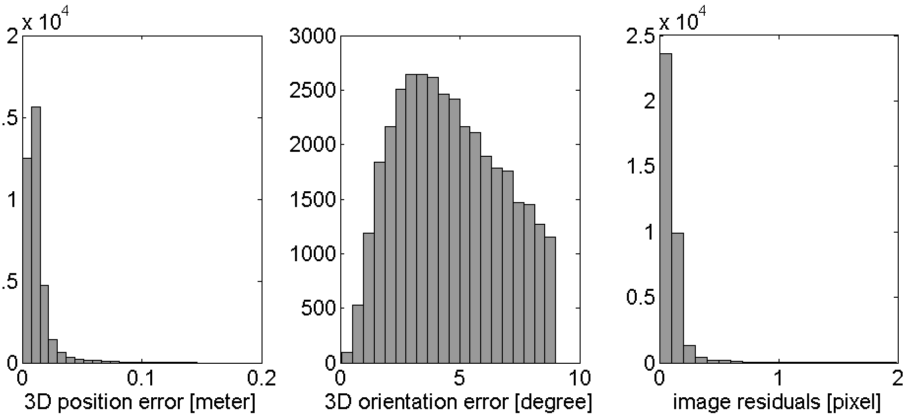


Figure 5.36: Histograms for the estimated position uncertainty (a) and the estimated orientation uncertainty (b). The estimates are computed using error propagation from estimated a priory uncertainties of the extracted 2D features. The mean position uncertainty is ±12.5mm and the mean orientation uncertainty is ±3.9 degrees. The histogram that illustrates the orientation uncertainty is cut at 9 degrees - this was the predefined threshold for accepting 3D hypothesis during the sweeping.

The second data set that is investigated consists of five images of the Mars statue at the main entry to the Landhaus courtyard in Graz. The scene differs significantly from the courtyard scenario - in this case the camera motion describes an approximate half circle around the object. Figure 5.37 shows three of the five images and renderings of the reconstructed 3D primitives from different viewpoints. The overall structure of the statue is captured and many details show up in the reconstruction. This scene is especially interesting due to the presence of many curved lines. The rendering shows that many curves were reconstructed correctly, particularly at examining the reprojections in figure 5.38.

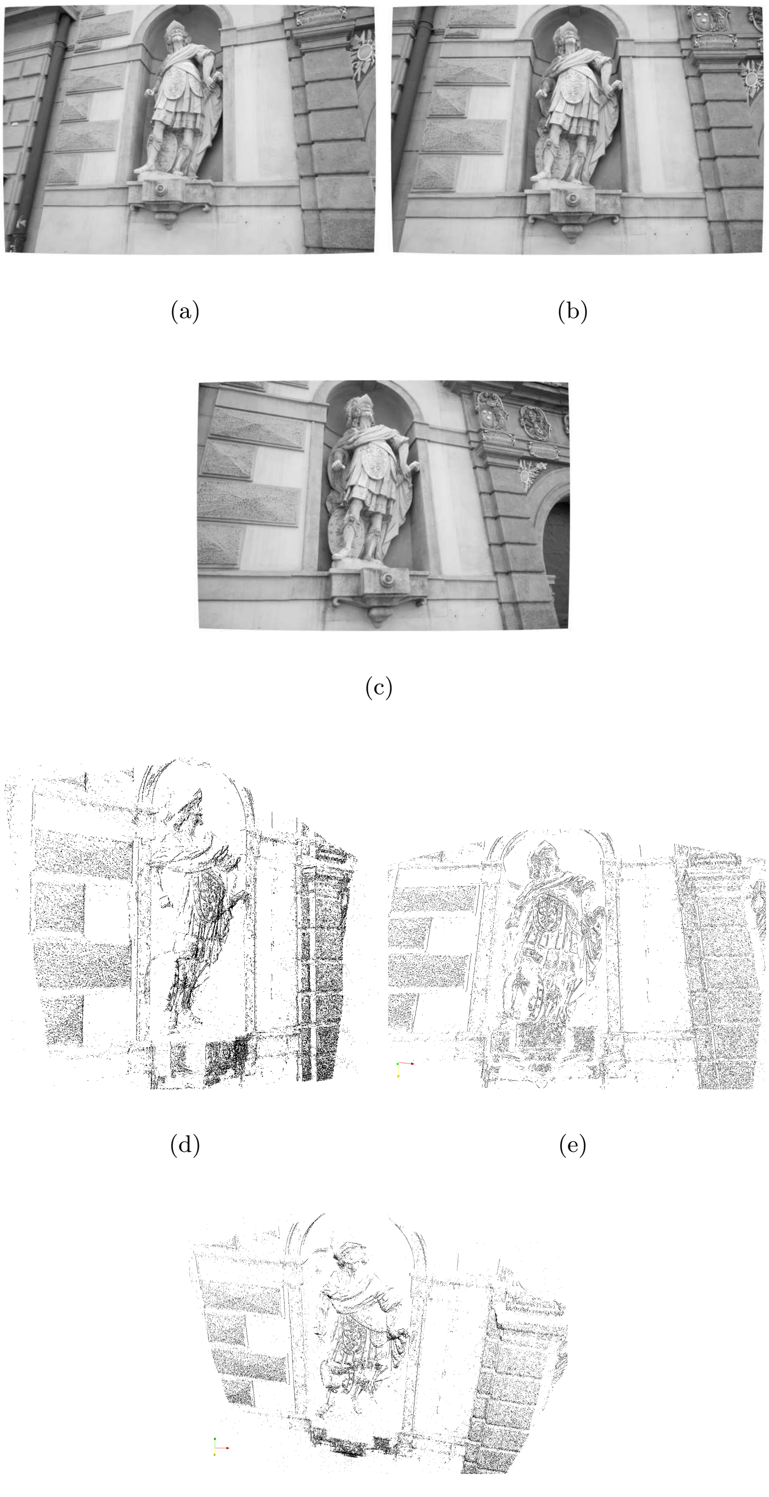
As for the courtyard scene overlays of edgels, and projected 3D primitives onto the source images are presented. The statues head and part of the upper body is shown in four such overlays in figure 5.38. A notably observation is that despite a significant change of viewpoints and thus significant differences in the appearance of the first and the last image in the sequence, many primitives are reconstructed correctly. One outlier is visible in the upper right corner of image 5.38(d).

As in the courtyard example the covariances for the position and orientation are computed by error propagation. The same a priori position uncertainty (±0.25 pixel) and orientation uncertainty (±3 degrees) are assumed. Figure 5.39 shows the histograms for the a posteriori uncertainties of the position and the orientation.

The mean position uncertainty is $\pm 5.7mm$ and the mean orientation uncertainty is ±3.6 degrees. The histogram that illustrates the orientation uncertainty is cut at 7 degrees - this was the predefined threshold for accepting 3D hypotheses during the sweeping. The fact that the position uncertainty is significantly lower for the statue scene as it was for the courtyard scene can be explained by the different camera motion: While the camera motion was dominantly linear for the courtyard scene, it was nearly circular for the statue scene. Another cause is the object distance - the statue was only 2 to 4 meters from the cameras, the distance from the facade to the camera positions was more than 20 meters.

These two experiments show the influence of the recording strategy on the final results. In traditional photogrammetric recording considerable effort goes into the mission planning. In this stage the best recording positions for a given scene are determined. The goal of this strategy is to minimize the number of images that are necessary for the desired measurements and to determine camera positions that allow the most accurate measurements. In an automated recording mission that is carried out with a mobile platform such considerations can often not be incorporated due to restricted access to the objects that are to be recorded or due to mechanical limitations of the recording platform itself. Automatic recording missions will therefore not be capable of producing recordings with the same fidelity as well planned manual recording missions. However this is compensated by increased redundancy. Mainly they will suffer from limitations that inhibit recording from the geometrically optimal view points and from the constraint that such platforms will have to be perpetually in motion.

In the experimental section we showed that the accuracy of the reconstructed 3D primitives is influenced by two factors: the accuracy of the 2D primitives and the configuration of the camera network. Since in an efficient city modeling work-flow the configuration of the camera network is constrained, an optimal system should ensure a vertical baseline between consecutive views. This allows to reconstruct structures that are parallel to the platforms motion vector. The magnitude of the vertical baseline will determine the achievable geometric accuracy for those structures. As a conclusion, it can be said that the design of a mobile recording platform will have a significant influence on the achievable geometric accuracy.



(a)

(b)

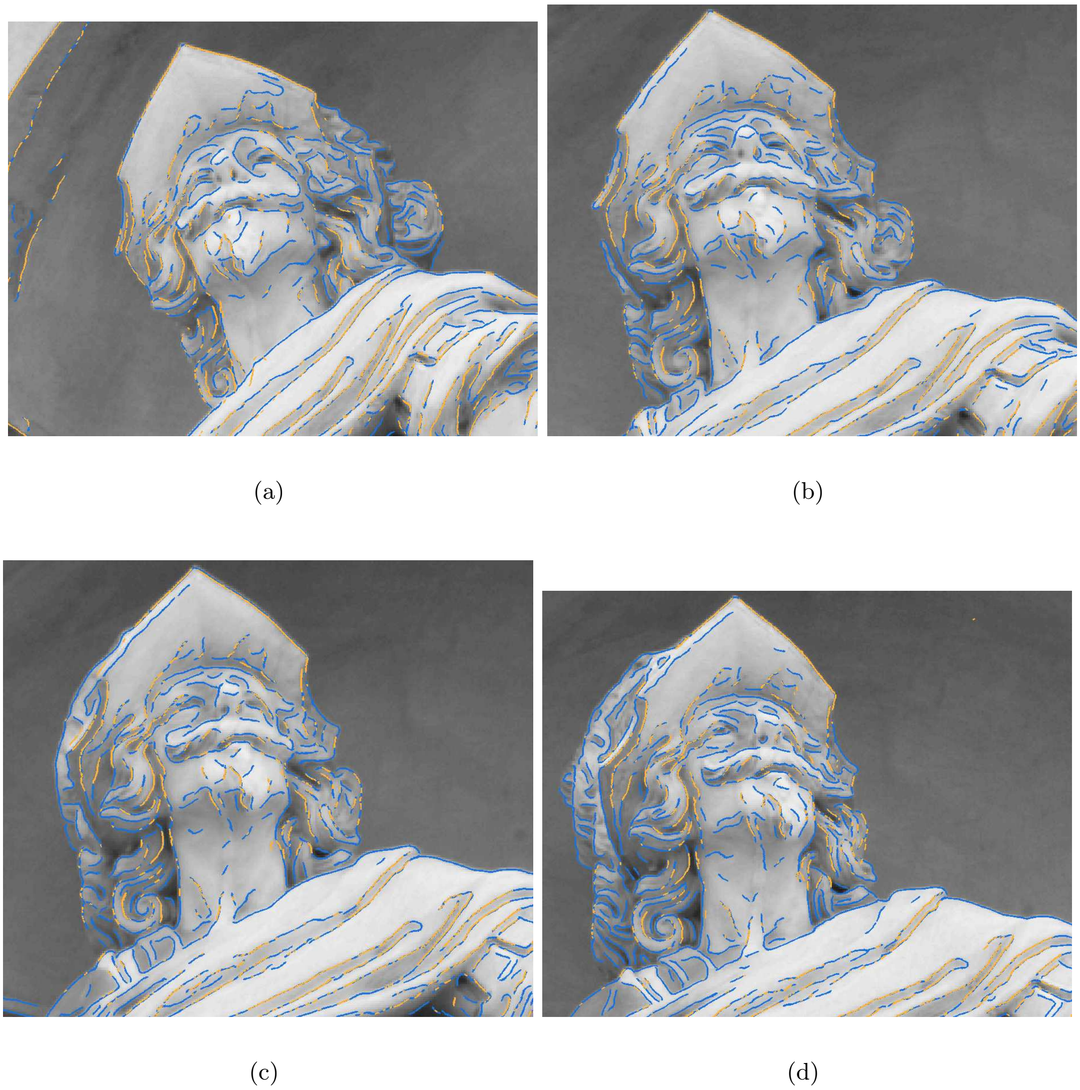
(c)

(d)

(e)

(f)

Figure 5.37: Top row: Three of the five statue images (image size = 2032 × 1352, 270k edgels on average). Bottom row: Three views of the resulting 3D point cloud.



(a)

(b)

(c)

(d)

Figure 5.38: Overlays of extracted 2D edgels (in blue) and projected 3D primitives onto four source images. This illustration gives an impression of the reconstruction accuracy by comparing the extracted 2D edgels position to the reprojected primitives position. Furthermore it shows that also for edgels that belong to curved lines a successful reconstruction can be performed. Also notable is the significant change of the images appearance due to large view point changes.

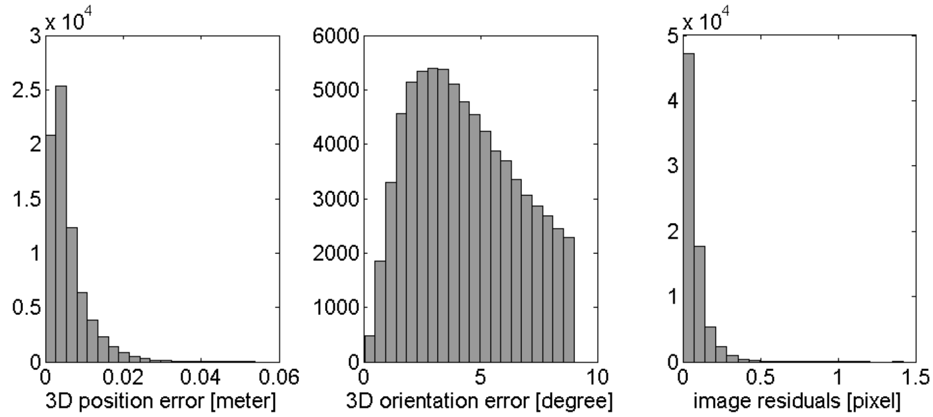


Figure 5.39: Histograms for the estimated position uncertainty (a) and the estimated orientation uncertainty (b). The estimates are computed using error propagation from estimated a priory uncertainties of the extracted 2D features. The mean position uncertainty is $0 \pm 5.7mm$ and the mean orientation uncertainty is ±3.6 degrees. The histogram that illustrates the orientation uncertainty is cut at 7 degrees - this was the predefined threshold for accepting 3D hypothesis during the sweeping.

Chapter 6

Summary and conclusion

Contents

This project is concerned with the fully automatic extraction of 3D primitives from digital images. The chosen testbed is the creation of 3D models of buildings as this is an area of ongoing research. The results contribute to the improvement of current methods used for modeling buildings from multiple images. The use of modern computer vision methods for the extraction of 3D data guarantees an efficient and robust way for data acquisition and furthermore allows for the fully automatic processing of large data sets. The consequent usage of image features throughout the whole processing chain showed the potential of the proposed methods to produce meaningful results.

Indeed, the project is a good starting point for anyone who is interested in feature based modeling. It presents powerful methods for extracting, processing geometric primitives and demonstrates a powerful 3D modeling approach.

6.1 Contributions of the thesis

The main contributions of the thesis can be divided into three major parts:

Accurate feature extraction In this part we showed that geometric features can be extracted in a robust and efficient manner. Especially the methods for computing points-of-interest from geometric features such as the Zwickel method demonstrated an alternative to standard approaches. The robust vanishing point extraction based on the Thales circle is an interesting new approach with applications in mobile mapping. The efficient detection of vanishing points allows for the estimation of the relative rotation (or components of the rotation) between successive images. The vertical vanishing provides a consistent estimate for the up-direction (which can be integrated as gauge constraint in a bundle adjustment approach). For two or more orthogonal vanishing points a rectification of scene planes can be performed. The rectified images are suited for automatic interpretation approaches. The robust fitting of affine squares has applications in the field of automatic semantic interpretation of images.

Robust correspondence estimation The proposed feature-based descriptors for matching points-of-interest were carefully derived for the particular case of city modeling, but can also be used in other application where similar image capturing strategies are used. In Urschler et al. [130] the use of the approximated shape context descriptor for the registration of volumetric computer tomography data sets was demonstrated.

Efficient 3D modeling by space sweeping The space sweep based 3D modeling approach, that works with chains of directed primitives is the main contribution of this project. This sparse modeling approach makes use of the high overlap between images within a sequence and is therefore well suited for the specific task of modeling buildings. We showed that the generation of directed 3D primitives can be achieved in an efficient manner and demonstrated a method for robust outlier removal based on energy minimization. We showed that high accuracy and efficiency do not contradict each other. In contrast to earlier feature-based 3D modeling approaches that work with sets of straight lines and conics, the ability to directly model from a set of arbitrary 2D chains produces significantly richer models. Applications for these models range from robust fitting of building models to direct primitive fitting for the individual 3D chains.

Using the proposed modeling method to tackle specific problems is also a possibility: The algorithm works well on problematic surfaces since it is inherently robust against outliers. A scenario could be the modeling of windows with specular reflections - dense image matching methods have problems finding the optimal surface whereas the proposed method only models the stationary contours. Another example is the modeling of cars in urban images. Cars are mainly responsible for large occlusions of the facades and their appearance (the surfaces are highly specular, smooth and textureless) makes a robust reconstruction with dense modeling nearly impossible, again the space sweep approach would only model the stationary outlines. In general we come to the conclusion that the proposed method constitutes a viable alternative to dense modeling techniques.

A point that can not be stressed enough, is the observation, that increases in robustness are mainly achieved by the high redundancy in the image sequences.

6.2 Open problems

This work merely scratched on the surface of the field. Nevertheless the proposed methods constitute improvements to existing methods. During the course of this work solutions to several problems originated, but in the same instant other problems emerged. In the following the most pressing problems from the author's view will be discussed and potential solutions will be proposed.

6.2.1 Data propagation

A mobile photogrammetric platform capturing its environment will produce image data in consecutive order and with a high redundancy (due to a high overlap between neighboring images). While this knowledge is exploited to speed up the tasks of image orientation it is almost neglected in the modeling stage. In the presented work the modeling was performed on small sets of images but for each set individually - no strategy for propagating 3D data to the next image set was performed. An improved strategy would be the sliding window approach that incrementally adds a new image and removes the oldest image of the sequence. Consequently a well designed strategy to propagate 3D primitives and then use these data for narrowing down the search space for potential matches could considerably lower the computational burden of the work flow. In many cases a new match might be directly used to improve the accuracy of an existing 3D primitive or the search for a match can be terminated prematurely. The necessity for making use of previously generated models is even higher in case of update missions.

6.2.2 Update missions

In order to keep a city model coherent, regular update missions are necessary. The automatic detection of significant changes and the completion of cartographic features such as roads or buildings can be achieved with higher certainty. In such missions the capturing platform is in charted territory and can make use of already existing information. An automatic location via image similarity measures and subsequent pose estimation can determine the exterior orientation parameters of the new images. If available, GPS coordinates can narrow down the search space. In urban environments robust change detection is an important topic and efficient 3D modeling methods can help in this context.

Another potential advantage of such missions is the possibility of completing geometry and textures for previously occluded areas (e.g. by cars on roads, vegetation, people, self-occlusions) and the refinement of existing models. Recognition methods can help by significantly reducing the complexity for the pose estimation by providing a rough localization through efficient delimiting of the number of possible camera poses. Through the introduction of further images the spatial accuracy of already reconstructed geometric primitives can be enhanced and ambiguities can be resolved. Guided reconstruction can be applied to densify regions where the reconstructed mesh has a low resolution and on the other hand abstraction algorithms can benefit from a denser modeled mesh. Radiometric properties of the imaged objects can be estimated if the recording missions are performed under different lighting conditions.

6.2.3 Segmentation vs. classification

The extraction of 3D information from images during the modeling stage relies on the segmentation of geometric primitives such as edgels, contour chains, line segments etc. This corresponds to a typical bottom-up scheme of data processing. Contrary to this approach a top-down scheme would perform a classification of the scene by semantically labeling regions in the images. This labelling partitions the scene into meaningful entities for which specific 3D reconstruction methods can be applied. Examples would be a detector for windows within the facades or a detector for cars.

6.2.4 Data fusion

The feature-based modeling and area-based reconstruction approaches produce 3D data of very different modalities. While feature-based 3D reconstruction methods produce sparse data with high spatial accuracy, area-based reconstruction methods produce dense height fields or triangle meshes with a reduced geometric accuracy in textureless regions and smoothed depth discontinuities. An ideal approach should join the 3D data produced by the two methods by making the best use of the somehow ”dual” characteristics of the data. The faster execution time of the feature-based approach implies that it should be ran in advance of any area-based method and then be used to initialize and guide the area-based modeling. An approach for using sparse point data in an area-based dense matching approach was demonstrated in [125].

6.2.5 Privacy

Whenever large urban areas are mapped the produced images contain people and the unedited publishing may therefore raise privacy concerns. In fact every non stationary object depicted in the imagery (especially people and cars) that may lead to the identification of a person, e.g. via the recognizable faces or license plates, are prone to misuse. It is not far-fetched to predict that after a few incidents of doubtful or unauthorized use of the data by a government agency an extensive public debate will take place. Even though in this thesis no method for producing texture information was proposed, the potential risks of misuse of high resolution terrestrial image data is worth mentioning. This line of argument holds since even the archiving of the raw data may constitute a violation of privacy laws. The huge effort of recent recording missions and produced amount of data will spawn efforts for the automatic blurring of faces and license plates in their wake.

6.2.6 Concluding remarks

During the work on this thesis many of the assumptions changed. While in 2001 the creation of 3D city models was mainly in the hand of a few specialized companies and academic institutions, it is now a battlefield for few (two?) global players. The reason for this is the need for constant maintenance of such fast changing entities as urban areas. Furthermore, a solid business model is needed to finance a globally spanning geo-information system. The near future will show to what degree this ubiquitous mapping will change our private lives. Maybe the next generation will develop a whole new sense of spatial embeddedness.

The integration of real time information into the visual presentation of geo-information will allow for fast assessment of complex local phenomena (e.g. real-time weather situation with cloud overlay, readings from various environmental sensors, real-time traffic information, GPS tracking for everybody but especially for children/property etc.). The amount of data that is provided by the general public is increasing exponentially. Especially digital images and videos are a valuable data source that is gaining more attention of the scientific community.

Some of the prospects may appear spooky at the first glance but the future will tell if they become socially accepted.

Bibliography

[1] AG, C. (2006). Cybercity modeler. `http://www.cybercity.tv`.

[2] Antone, M. E. and Teller, S. J. (2000). Automatic recovery of relative camera rotations for urban scenes. In *CVPR*, pages 2282–2289.

[3] Antone, M. E. and Teller, S. J. (2001). Scalable, absolute position recovery for omni-directional image networks. In *CVPR (1)*, pages 398–405.

[4] Baillard, C., Schmid, C., Zisserman, A., and Fitzgibbon, A. (1999a). Automatic line matching and 3D reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, IAPRS Vol.32, Part 3-2W5*, pages 69–80.

[5] Baillard, C., Schmid, C., Zisserman, A., and Fitzgibbon, A. (1999b). Automatic line matching and 3d reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, IAPRS Vol.32, Part 3-2W5*, pages 69–80.

[6] Baillard, C. and Zisserman, A. (2000). A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. In *19th ISPRS Congress and Exhibition*, Amsterdam.

[7] Bartoli, A. and Sturm, P. (2003a). Multiple-view structure and motion from line correspondences. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 207–212.

[8] Bartoli, A. and Sturm, P. (2005). Structure-from-motion using lines: representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100(3):416–441.

[9] Bartoli, A. and Sturm, P. F. (2003b). Multiple-view structure and motion from line correspondences. In *Proccedings of the 9th IEEE International Conference on Computer Vision (ICCV 2003)*, pages 207–212. IEEE Computer Society.

[10] Bauer, J., Bischof, H., Klaus, A., and Karner, K. (2004). Robust and fully automated image registration using invariant features. In *Proceedings of ISPRS - Int. Society for Photogrammetry and Remote Sensing*, pages 12–23.

[11] Bauer, J., Klaus, A., Karner, K., Zach, C., and Schindler, K. (2002). Metropogis: A feature based city modeling system. In *ISPRS Comission III Symposium, Graz, Conf. Proceedings*, volume B, pages 22–27.

[12] Bauer, J., Klaus, A., Sormann, M., , and Karner, K. (2005). Efficient 3d reconstruction by edgel sweeping. In *Proceedings of Optical3D (Optical 3-D Measurement Techniques)*, pages 253–262.

[13] Bauer, J., Zach, C., and Bischof, H. (2006). Efficient sparse 3d reconstruction by space sweeping. *3D Data Processing Visualization and Transmission, International Symposium on*, 0:527–534.

[14] Baumberg, A. (2000). Reliable feature matching across widely separated views.

[15] Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., and Ebner, H. (1999). Automatic road extraction based on multi-scale, grouping, and context. *Photogrammetric Engineering & Remote Sensing*, 65(7):777–785.

[16] Beaudet, P. (1978). Rotationally invariant image operator. In *Proceedings of the Fourth International Conference on Pattern Recognition*, pages 579–583.

[17] Belongie, S., Malik, J., and Puzicha, J. (2000). Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837.

[18] Berg, A. C. and Malik, J. (2001). Geometric blur for template matching. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, volume 1, pages 607–614.

[19] Bleyer, M. and Gelautz, M. (January 2005). Graph-based surface reconstruction from stereo pairs using image segmentation. In *SPIE*, pages vol. 5665: 288–299.

[20] Borgefors, G. (1983). Chamfering: A fast method for obtaining approximations of the Euclidean distance in N dimensions. In *Proc. 3rd Scand. Conf. on Image Analysis (SCIA3)*, pages 250–255, Copenhagen, Denmark.

[21] Brauer-Burchardt, C. and Voss, K. (2000). Robust vanishing point determination in noisy images. In *15th International Conference on Pattern Recognition (ICPR'00)*, volume 1, pages 559–562.

[22] Brown, M. and Lowe, D. G. (2002). Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference*, pages 656–665.

[23] Burns, J. B., Hanson, A. R., and Riseman, E. M. (1986). Extracting straight lines. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(4):425–455.

[24] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679-698.*, 8(6):679–698.

[25] Cheng, Y. and Lee, S. (1995). A new method for quadratic curve detection using k-ransac with acceleration techniques. *Pattern Recognition*, 28:663–682.

[26] Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *Image Understanding Workshop*, pages 1213–1220.

[27] Collins, R. T. and Weiss, R. S. (1990). Vanishing point calculation as a statistical inference on the unit sphere. In *International Conference on Computer Vision*, pages 400–403.

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

[29] Coorg, S. and Teller, S. (1999). Extracting textured vertical facades from controlled close-range imagery. In *CVPR*, pages 625–632.

[30] Deng, Y., Yang, Q., Lin, X., and Tang, X. (2005). A symmetric patch-based correspondence model for occlusion handling. In *ICCV*, pages II: 1316–1322.

[31] Devernay, F., Devernay, F., Robotique, P., and Robotvis, P. (1995). A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical report, INRIA Research Report 2724, SophiaAntipolis.

[32] Engeln-Müllges, G. and Uhlig, F. (1996). *Numerical Algorithms with C*. Springer, Berlin, 1 edition.

[33] Fischler, M. and Bolles, R. (1981). Ransac random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 26:381–395.

[34] Fitzgibbon, A. and Fisher, R. (1995). A buyer's guide to conic fitting.

[35] Fitzgibbon, A., Pilu, M., and Fisher, R. B. (1999). Direct least square fitting of ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):476–480.

[36] Fitzgibbon, A. W. (2001). Robust registration of 2d and 3d point sets. In *Proceedings, British Machine Vision Conference*.

[37] Folkesson, J., Jensfelt, P., and Christensen, H. (2005). Vision SLAM in the measurement subspace. In *Intl Conf. on Robotics and Automation*, pages 30 – 35, Barcelona, ES. IEEE.

[38] Forsyth, D. A. and Ponce, J. (2003). *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 1 edition.

[39] Fritz, G., Seifert, C., and Paletta, L. (2006). A mobile vision system for urban detection with informative local descriptors. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 131 – 137.

[40] Förstner, W. and Gülch, E. (1987). A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *In Proceedings of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data, Interlaken, Switzerland*, pages 281–305.

[41] Früh, C. and Zakhor, A. (2004). An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60(1):5–24.

[42] Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22.

[43] Garland, M. and Heckbert, P. (1997). Surface simplification using quadric error metrics. *Proceedings of SIGGRAPH'97*, pages 209–215.

[44] Goedeme, T., Tuytelaars, T., Vanacker, G., Nuttin, M., and Gool, L. V. (2005). Feature based omnidirectional sparse visual path following. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1003–1008.

[45] Gonzalez, R. and Woods, R. (2002). *Digital image processing*. Prentice Hall, Upper Saddle River, New Jersey, 2 edition.

[46] Google$^{\mathrm{TM}}$(2007). Google streetview. `http://www.maps.google.com/help/maps/streetview`.

[47] Grimm, A. (2007). The origin of the term photogrammetry. In *Proceedings of the Photogrammetric Week '07*, pages 53–60.

[48] Haala, N., Peter, M., Kremer, J., and Hunter, G. (2008). Mobile lidar mapping for 3d point cloud collection in urban areas: A performance test. In *Proceedings of the XXI ISPRS Congress, July 2008 Beijing, CHINA*, page 1119 ff.

[49] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings 4th Alvey Visual Conference*, pages 147 – 151.

[50] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[51] Heuel, S. and Förstner, W. (2001). Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages 517–524. IEEE Computer Society.

[52] Hong, L. and Chen, G. (2004). Segment-based stereo matching using graph cuts. In *CVPR*, pages I: 74–81.

[53] Hoppe, H. (1996). Progressive meshes. *Proceedings of SIGGRAPH '96*, pages 99–108.

[54] Hough, P. (1962). Method and means for recognizing complex patterns. US patent 3,069,654.

[55] Hu, J., You, S., and Neumann, U. (2003). Approaches to large-scale urban modeling. *IEEE Computer Graphics and Applications*, 23(6):62–69.

[56] Huet, B. and Hancock, E. (1996). Cartographic indexing into a database of remotely sensed images. In *WACV'96, pages 8–14, Dec 1996.*

[57] Illingworth, J. and Kittler, J. (1988). A survey of the hough transform. *Computer Vision, Graphics and Image Processing*, 44:87–116.

[58] Inc., E. S. (2006). Photomodeler. `http://www.photomodeler.com/`.

[59] J., B., A., K., M., S., and K., K. (2004). Sparse 3d reconstruction by edgel sweeping. In *Proceedings of the CVWW - Computer Vision Winter Workshop*, pages 11–20.

[60] J. Bauer, K. K. and Schindler, K. (2002). Plane parameter estimation by edge set matching. In *Proceedings of the 26th Workshop of the Austrian Association for Pattern Recognition*, pages 29–36.

[61] J. Matas, O. Chum, U. M. and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *In Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393.

[62] Jaynes, C., Riseman, E., and Hanson, A. (2003). Recognition and reconstruction of buildings from multiple aerial images. *Computer Vision and Image Understanding*, 90(1):68–98.

[63] Jiang, W., Er, G., Dai, Q., and Gu, J. (2006). Similarity-based online feature selection in content-based image retrieval. *IEEE Trans. Image Processing*, 15(3):702–712.

[64] Jung, F., Tollu, V., and Paparoditis, N. (2002). Extracting 3d edgels hypotheses from multiple calibrated images: A step towards the reconstruction of curved and straight object boundary lines. *ISPRS Journal of Photogrammetric Computer Vision*, B:100104.

[65] Kang, S., Szeliski, R., and Chai, J. (2001). Handling occlusions in dense multiview stereo. In *In IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, December 2001*, volume 1, pages 103–110.

[66] Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

[67] Kim, Z. and Nevatia, R. (2004). Automatic description of complex buildings from multiple images. *Computer Vision and Image Understanding*, 96(1):60–95.

[68] Koch, R., Pollefeys, M., and Gool, L. V. (2000). Realistic surface reconstruction of 3d scenes from uncalibrated image sequences. *Journal of Visualization and Computer Animation*, 11:115–127.

[69] Kolmogorov, V. and Zabih, R. (2002). Multi-camera scene reconstruction via graph cuts. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 82–96, London, UK. Springer-Verlag.

[70] Kosecká, J. and Zhang, W. (2002). Efficient computation of vanishing points. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 223–228.

[71] Kosecká, J. and Zhang, W. (2005). Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):274–293.

[72] Kutulakos, K. N. and Seitz, S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision*, 38:199–218.

[73] Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(8):1265–1278.

[74] Leberl, F. and Gruber, M. (2003). Large format aerial digital camera - aerial film photogrammetry coming to end. *GIM International, The worldwide Magazine for Geomatics*, 17(6).

[75] Leberl, F. and Gruber, M. (2005). ULTRACAM-D: Understanding some noteworthy capabilities. In Fritsch, D., editor, *Photogrammetric Week '05*, pages 57–68. Wichmann Verlag, Heidelberg.

[76] Lourakis, M. and Argyros, A. (2004). The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece. Available from `http://www.ics.forth.gr/~lourakis/sba`.

[77] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157.

[78] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

[79] Martinec, D. and Pajdla, T. (2005). 3d reconstruction by fitting low-rank matrices with missing data. In *CVPR (1)*, pages 198–205.

[80] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proc. 13th British Machine Vision Conference, Cardiff, UK*, pages 384–393.

[81] Micusík, B., Wildenauer, H., and Kosecka, J. (2008). Detection and matching of rectilinear structures. In *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

[82] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *Proceedings of the International Conference on Computer Vision, Vancouver, Canada*, pages 525–531.

[83] Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *European Conference on Computer Vision*, volume I, pages 128–142.

[84] Mikolajczyk, K. and Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[85] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630.

[86] Mikolajczyk, K. and Schmid, C. (June 2003). A performance evaluation of local descriptors. In *International Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, volume 2, pages 257–263.

[87] Miled, W. and Pesquet, J. C. (2006). Disparity map estimation using a total variation bound. *Computer and Robot Vision, Canadian Conference*, 0:48.

[88] Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 584–587.

[89] Mortensen, E., D., D. H., and Shapiro, L. (2005). A sift descriptor with global context. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 184 – 190.

[90] Murale, D. (2007). 3d measurement and virtual reconstruction of ancient lost worlds of europe. `http://www.dea.brunel.ac.uk/project/murale`.

[91] Nelson, R. C. (1994). Finding line segments by stick growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):519–523.

[92] Osian, M., Tuytelaars, T., and Gool, L. V. (2004). Fitting superellipses to incomplete contours. *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, 4:49–57.

[93] Pilu, M., Fitzgibbon, A., and Fisher, R. (1996). Ellipse-specific direct least-square fitting.

[94] Pilu, M., Fitzgibbon, A., and Fisher, R. (1999). Direct least-squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480.

[95] Pollefeys, M., Gool, L. V., and Proesmans, M. (1996). Euclidean 3d reconstruction from image sequences with variable focal lengths. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, pages 31–42.

[96] Pollefeys, M., Koch, R., Vergauwen, M., and Gool, L. V. (1998). Flexible 3d acquisition with a monocular camera. In *In Proceedings of the IEEE Int'l Conf. on Robotics and Automation*, pages 2771–2776.

[97] Pritchett, P. and Zisserman, A. (1998a). Matching and reconstruction from widely separated views. In Koch, R. and Van Gool, L., editors, *3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506*, pages 78–92. Springer.

[98] Pritchett, P. and Zisserman, A. (1998b). Wide baseline stereo matching. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pages 754–760.

[99] Quan, L. (1996). Conic reconstruction and correspondence from two views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):151–160.

[100] REALVIZ (2006). Imagemodeler. `http://www.realviz.com/`.

[101] Riegl (2007). Lms z390 terrestrial laser scanner. `http://www.riegl.co.at/terrestrial_scanners/lms-z390_/390_all.htm`.

[102] Rosin, P. L. (2000). Fitting superellipses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7):726–732.

[103] Rother, C. (2000). A new approach for vanishing point detection in architectural environments. In *BMVC*, pages 382–391.

[104] Rother, C. (2003). Linear multi-view reconstruction of points, lines, planes and cameras using a reference plane. In *Proceedings of the ICCV*, pages 1210–1217.

[105] Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2003). 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *CVPR (2)*, pages 272–280.

[106] Rothwell, C., Mundy, J., Hoffman, W., and Nguyen, V. (1995). Driving vision by topology. In *In Proceedings IEEE Symposium on Computer Vision SCV95*, pages 395–400.

[107] Rubner, Y., Tomasi, C., and Guibas, L. J. (1998). A metric for distributions with applications to image databases. In *Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India, January 1998*, pages 59–66.

[108] Schaffalitzky, F. and Zisserman, A. (2000). Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18(9):647–658.

[109] Schaffalitzky, F. and Zisserman, A. (2001). Viewpoint invariant texture matching and wide baseline stereo. In *Proc. 8th International Conference on Computer Vision, Vancouver, Canada*.

[110] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535.

[111] Schmid, C. and Zisserman, A. (1997). Automatic line matching across views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671.

[112] Schmid, C. and Zisserman, A. (2000a). The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233.

[113] Schmid, C. and Zisserman, A. (2000b). The geometry and matching of lines and curves over multiple views. *IJCV*, 40(3):199–233.

[114] Se, S., Lowe, D. G., and Little, J. J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375.

[115] Seitz, S. M. and Dyer, C. R. (1997). Photorealistic scene reconstruction by voxel coloring. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1067–1073.

[116] Shahrabi, A. (2000). *Automatic recognition and 3D reconstruction of buildings from digital imagery*. Beck.

[117] Shlyakhter, I., Rozenoer, M., Dorsey, J., and Teller, S. J. (2001). Reconstructing 3D tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 21(3):53–61.

[118] Sturm, P. and Triggs, B. (1996). A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision*, pages 709–20, Cambridge, U.K. Springer-Verlag.

[119] Sun, J., Li, Y., Kang, S., and Shum, H. (2005). Symmetric stereo matching for occlusion handling. In *CVPR*, pages II: 399–406.

[120] Systems, G. (2006). Geosim systems. `http://www.geosimcities.com`.

[121] T., L. (1998). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–154.

[122] T., L., S., R., S., K., and I., H. (2006). *Close Range Photogrammetry; Principles; Methods and Applications*. Whittles, first edition.

[123] Tell, D. and Carlsson, S. (2002). Combining appearance and topology for wide baseline matching. In *European Conference on Computer Vision*, volume 1, pages 68–81.

[124] Teller, S. J., Antone, M. E., Bodnar, Z., Bosse, M., Coorg, S. R., Jethwa, M., and Master, N. (2003). Calibrated, registered images of an extended urban area. *International Journal of Computer Vision*, 53(1):93–107.

[125] Torr, P. and Criminisi, A. (2004). Dense stereo using pivoted dynamic programming. *Image and Vision Computing*, 22(10):795–806.

[126] Tuytelaars, T. and Gool, L. J. V. (2004). Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85.

[127] Tuytelaars, T. and Gool, L. V. (2000). Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference BMVC'2000*.

[128] Tuytelaars, T., Proesmans, M., and Gool, L. J. V. (1997). The cascaded hough transform as support for grouping and finding vanishing points and lines. In *AFPAC '97: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle*, pages 278–289, London, UK. Springer-Verlag.

[129] Ulm, K. (2003). Improved 3D city modeling with cybercity-modeler (CC-Modeler^{TM}) using aerial-, satellite imagery and laserscanner data. In *Proceedings of the ISPRS Workshop on Visualization and Animation of Reality-based 3D Models*, pages 87–92.

[130] Urschler, M., Bauer, J., Ditt, H., and Bischof, H. (2006). Sift and shape context for feature-based nonlinear registration of thoracic ct images. In *Proceedings of the 2nd International ECCV Workshop on Computer Vision Approaches to Medical Image Analysis*, pages 73–84.

[131] van den Heuvel, F. A. (1998). Vanishing point detection for architectural photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, 32(5):652–659.

[132] Venkateswar, V. and Chellappa, R. (1992). Extraction of straight lines in aerial images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(11):1111–1114.

[133] Vogel, J. and Schiele, B. (2002). Query-dependent performance optimization for vocabulary-supported image retrieval. In *German Pattern Recognition Symposium*, page 600 ff.

[134] Werner, T., Schaffalitzky, F., and Zisserman, A. (2001). Automated architecture reconstruction from close-range photogrammetry. In *Proc. on CIPA 2001 International Symposium: Surveying and Documentation of Historic Buildings – Monuments – Sites, Traditional and Modern Methods*.

[135] Werner, T. and Zisserman, A. (2002). New techniques for automated architecture reconstruction from photographs. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 2, pages 541–555. Springer.

[136] X., Z. and P.L., R. (2003). Superellipse fitting to partial data. *Pattern Recognition*, 36(3):743–752.

[137] Yáng, Q., Wang, L., Yang, R., Stewénius, H., and Nistér, D. (2006). Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *Proceedings of the CVPR*, volume 2, pages 2347 – 2354.

[138] Zach, C., Pock, T., and Bischof, H. (2007). A globally optimal algorithm for robust tv-l1 range image integration. In *Proceedings of the International Conference on Computer Vision*, pages 1–8.

[139] Zhang, X. and Rosin, P. L. (2003). Superellipse fitting to partial data. *Pattern Recognition*, 36(3):743–752.

[140] Zisserman, A., Werner, T., and Schaffalitzky, F. (2001). Towards automated reconstruction of architectural scenes from multiple images. In *Proc. 25th workshop of the Austrian Association for Pattern Recognition*, pages 9–23.