Dorota Iwaszczuk

Automatic texturing of 3D models of urban areas
using image sequences from airborne TIR cameras

München 2015

Automatic texturing of 3D models of urban areas using image sequences from airborne TIR cameras

Vollständiger Abdruck

der von der Ingenieurfakultät Bau Geo Umwelt

der Technischen Universität München

zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation

von

Dorota Iwaszczuk

München 2015

Prüfungskommission

Vorsitzender:	Univ.-Prof. Dr.-Ing. Liqiu Meng
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. Uwe Stilla
	2. Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe
	3. Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Wolfgang Förstner, Rheinische Friedrich-Wilhelms-Universität Bonn

Die Dissertation wurde am 23.02.2015 bei der Technischen Universität München eingereicht
und durch die Ingenieurfakultät Bau Geo Umwelt am 18.05.2015 angenommen.

Abstract

Thermal infrared (TIR) images are often used to picture damaged and weak spots in the insulation of the building hull, which is widely used in thermal inspections of buildings. The goal of this study is to provide a groundwork for such inspection in large-scale areas in the form of geo-referenced TIR textures for 3D building models. This avoids time consuming imaging and manually analyzing each face independently. It also enables the extraction of façade structures so that they can be embedded together with the TIR textures in a 3D spatial information system.

For this purpose, 3D building models and TIR images need to be co-registered. Direct geo-referencing as a co-registration method is usually not sufficient for precise texture extraction. Hence, an appropriate model-to-image matching is required. The majority of the existing solutions for model-to-image matching do not take the errors and uncertainties of the 3D models into account. Usually, textures are extracted for triangulated models and existing methods do not consider representation based on arbitrary polygons. Moreover, only few researchers assess the quality of extracted textures, but even they fail to pay attention to the quality of the fit between the 3D building models and the textures. Most of the methods for façade reconstruction are based on classification or learning and require large database of training data, which is not available for TIR images. Almost all the methods are designed for the visible domain and do not investigate other spectral bands.

In this work, methods and strategies for precise texture extraction from airborne TIR image sequences are developed, and the potential for windows detection in the extracted textures is evaluated. In order to capture all faces, including the roofs, façades, and façades in the inner courtyard, an oblique looking video camera mounted on a flying platform is used. For this acquisition configuration, methods for a line-based model-to-image matching are developed, which consider uncertainties of the 3D building model, as well as of the image features, and determine the optimal exterior parameters of the camera. The remaining geometric mismatch between the projected 3D building model and image structures is compensated for every texture locally. This is done by adjusting the projected edges of the 3D building model to the gradient image generated from the TIR image. Moreover, this study investigates whether line tracking through the image sequence supports the matching. The accuracy of the extraction and the quality of the textures are assessed. Finally, a method for window detection in thermal textures based on lattice reconstruction is presented.

The tests showed good results on co-registration, particularly in cases where tracking between the neighboring frames had been applied. Local matching also yielded improvement of the fit between the image features and the model edges, which enabled the extraction of better fitting textures. The extracted textures, however, are of low resolution and contrast, which makes it difficult to use them for window detection.

Kurzfassung

Bei der thermischen Gebäudeinspektionen kommen häufig Kameras zum Einsatz, die Bilder im Thermischen Infrarot (TIR) aufnehmen, um Schad- oder Schwachstellen bei der Isolation von Gebäudehüllen sichtbar zu machen. Ziel der Arbeit ist es, für großflächige urbane Bereiche solche Inspektionen zu ermöglichen. Um zeitintensive manuelle Auswertungen zu vermeiden, sollen georeferenzierte TIR-Texturen für 3D Gebäudemodelle automatisch abgeleitet und gespeichert werden. Zusätzlich können Fassadenstrukturen bezüglich der Geometrie und Radiometrie ausgewertet werden und in einem 3D Informationssystem (GIS, BIM) einzubinden.

Dafür müssen die 3D Gebäudemodelle und TIR Bilder koregistriert werden. Eine direkte Georeferenzierung ist für die Koregistrierung meistens nicht ausreichend um Gebäudetexturen präzise zu extrahieren. Daher muss eine geeignete Zuordnung zwischen Model und Bild implementiert werden. Die meisten bekannten Zuordnungsmethoden berücksichtigen keine Fehler und Ungenauigkeiten des 3D-Gebäudemodells. Die Texturen werden üblicherweise für triangulierte Modelle extrahiert. Somit können als beliebige Polygone gespeicherten Gebäudemodelle nicht behandelt werden. Nur wenige Autoren bewerten die Qualität der extrahierten Texturen, keine Arbeit jedoch die Qualität der Übereinstimmung zwischen den Gebäudemodellen und den Texturen. Die meisten Methoden zur Fassadenrekonstruktion basieren auf Klassifizierung oder maschinellem Lernen und benötigen umfangreiche Trainingsdaten, die für TIR Bilder nicht verfügbar sind. Außerdem sind die meisten Methoden für Bilder in sichtbaren Wellenlängen ausgelegt und befassen sich nicht mit anderen Spektralbereichen wie Infrarot.

In dieser Arbeit werden Methoden und Strategien für eine präzise Texturextraktion aus TIR Bildsequenzen entwickelt. Zudem wird das Potential, aus diesen Texturen Fenster zu detektieren, evaluiert. Um sämtliche Flächen der Gebäudehülle erfassen zu können, wird eine flugzeuggetragene Kamera in Schrägsichtperspektive verwendet. Für diese Aufnahmekonfiguration werden Methoden für eine linienbasierte Zuordnung zwischen Model und Bild entwickelt welche Fehler und Ungenauigkeiten der 3D-Gebäudemodelle sowie der Bildmerkmale berücksichtigen und die Berechnung der optimalen Parameter der äußeren Orientierung der Kamera ermöglichen. Die nach der Optimierung verbleibende geometrische Diskrepanz zwischen Gebäudemodell und TIR Bild wird für jede Textur einzeln lokal kompensiert. Dies wird durch Anpassung von Kanten des projizierten 3D-Gebäudemodells an ein aus dem TIR Bild generieten Gradientenbild erreicht. Zudem wird untersucht, ob eine Linienverfolgung durch die Bildsequenz die Zuordnung unterstützt. Die Genauigkeit der Extraktion und die Qualität der Texturen werden bewertet. Schließlich wird ein Verfahren zur Erkennung von Fenstern in TIR Texturen durch Gitter-Rekonstruktion vorgestellt.

Die Versuche zeigen gute Resultate für die Koregistrierung, insbesondere wenn eine Linienverfolgung zwischen zwei benachbarten Bildern durchgeführt wurde. Das lokale Matching-Verfahren führt zu verbesserter Übereinstimmung der Bildmerkmale mit den Modellkanten was wiederum die Extraktion von gut angepassten Texturen ermöglicht. Jedoch sind die Auflösung und der Kontrast der extrahierten TIR Texturen gering, was die Fensterdetektion erschwert.

Contents

1 Introduction

1.1 Motivation

The amount of energy used by buildings is a significant part of the total energy consumed by humans. Research has shown that in European countries buildings consume 40% [Baden et al., 2006] of all produced energy. Of the energy consumed in buildings, 47% is used for heating purposes [Hennigan, 2011]. Due to climate change, increasing energy costs, and energy performance directives, the energy efficiency of buildings should be improved. Recent technologies allow new buildings to be constructed with higher energy performances, but older buildings still need to be inspected in order to improve their energy efficiency.

Thermal inspections of buildings using a thermal infrared (TIR) camera contribute to the detection of damaged and weak spots in building structures [Balaras & Argiriou, 2002]. Thermal cameras capture the heat radiation of the building hull and record it as an intensity image. The intensity value corresponds to a remotely measured temperature on a defined scale. Nowadays, thermal cameras can achieve high radiometric resolution of up to 0.01 [K], which means that it is possible to capture even very small temperature differences. Thanks to this, we can detect heat leakages in thermal images of a building hull. The knowledge of the location of heat leaks is used for renovation planning or for quality control after the renovation. Currently, most thermal inspections of buildings use a single TIR image analysis, without reference to the geometry of the captured scene.

The TIR images are often taken manually for a single building. However, we can also observe a trend to apply mobile mapping systems to carry out large scale inspections for urban quarters [Hoegner et al., 2007] or entire cities [Chandler, 2011]. For this purpose, the TIR cameras are mounted on a mobile terrestrial platform, for example on a vehicle. Terrestrial TIR images are used to documentat frontal faces (façades) visible from the street level, but they do not capture roofs and inner yards. Roofs can be seen from a flying platform, such as an unmanned aerial vehicle (UAV) or a helicopter. Using oblique view images, inner yards can also be captured. Combining TIR images with three-dimensional (3D) geometries allows for the spatial reference of the thermal data and facilitates their interpretation. Thermal data can be combined with different kinds of 3D geometries: Building Information Models (BIM) [Mikeleit & Kraneis, 2010], 3D building models via texture mapping [Hoegner et al., 2007; Iwaszczuk et al., 2011c], 3D point clouds via assignment and interpolation of the measured temperature to the points [Cabrelles et al., 2009; Borrmann et al., 2012; Vidas et al., 2013] or aerial photographs combined with a point cloud [Boyd, 2013]. Using a point cloud as spatial reference enables fast generation of results with a high level of detail and is appropriate for visual interpretation, while 3D building models deliver more generalized and structured representations to support automatic analysis. Embedding thermal data in a geo-database [Kumke et al., 2006] allows for spatial queries and analysis.

For thermal inspections, a variety of data types are valuable. On the one hand, information extracted from thermal imagery, such as thermal textures, detected heat leakages, windows,

and other façade structures, should be stored and managed in a database together with the 3D geometries. On the other hand, information extracted from other sources, such as remote sensing data and spatial information systems, are also very important. Information about material of the photographed object plays an important role in the physical interpretation of thermal images. This information can be extracted using multi-spectral or hyper-spectral data, stored in a database and used for analysis. Other information including building's address, owner, type, function, usage, construction year, solar collectors, and further attributes and properties can also be useful. Nowadays, this information becomes successively a norm in Geographic Information Systems (GIS), as can be seen by the developement of new standards, such as City Geography Markup Language (*CityGML*)[Gröger et al., 2012]. This standardization enables interoperability of the data and facilitates data access and exchange [Kolbe et al., 2005].

1.2   Objectives of the Work

The goal of this work is to provide the groundwork for large area thermal inspections of buildings in form of thermal infrared textures for 3D building models. Based on these textures, an assessment of the thermal efficiency of buildings, building complexes, districts, or entire cities is expected to be possible. Particularly, the detection and geo-location of thermal leakages in the building hull should be enabled. The extracted TIR textures, therefore, are required to be the best possible quality in terms of fit and level of detail. A further requirement is to capture, if possible, the entire building hull, especially roofs and inner yard façades, which can not be captured from the street view level. Hence, in this work, methods and strategies will be developed and tested that enable the extraction of well fitting textures of high geometric quality.

For this purpose, the 3D building model has to be transformed in one coordinate system with the thermal images. This process is called *"co-registration"*. An algorithm for selection and assignment of image regions to the model polygons will also be presented. This process is called *"texture extraction"*. The usability of the extracted textures for *extraction of "façade structures"*, such as windows, doors, and heat leakages, will be assessed based on the first attempt to interpret thermal textures in terms of detecting windows.

In order to best capture all faces (roofs and walls) of a 3D building model, an appropriate acquisition geometry is needed. This will be achieved by capturing the scene from a flying platform using an oblique looking camera. Additionally, the camera should be able to capture the scene with a high frame rate to ensure that as many faces as possible are taken from an advantageous position and viewing angle. This will result in a large overlap between frames.

Accordingly, the methods presented in this work are developed primarily for oblique view thermal infrared image sequences. For this thesis, four main objectives are defined and listed, and they are ordered according to the focus of the thesis.

In an attempt to extract the best fitting textures, directly geo-referenced thermal images will be first co-registered with an existing 3D building model, which is the first objective and main focus of this work. The second objective is extracting textures with respect to the texture quality and their automatic assignment to the 3D building model. The third objective is to define the quality measures enabling the assessment of each single texture and whole texture extraction process, with respect to the level of detail of textures and the quality of co-registration. The fourth and last objective is detecting windows and embedding them in the 3D building model.

1.3 Structure of the Work

This thesis contains ten chapters.

Chapter 1 introduces the topic, defines the objectives of the study, and presents the structure of the work.

Chapter 2 presents current research on co-registration, texture extraction and building enrichment by textures and façade structures. The strength and weakness of the existing approaches are discussed and a new strategy for each of these topics is proposed.

Chapter 3 explains the theoretical background which is used in Chapters 4-6. First, the basics of thermal imaging and properties of thermal imagery are explained. Then, selected mathematical foundations in photogrammetry, image processing, and statistics are presented. This description has a great impact in Chapters 4-6 because it provides the mathematical background used in the methodology presented in this chapter.

Chapter 4 addresses the coregistration of 3D building models with TIR imagery. In coregistration, the flight trajectory of the sensor is estimated in order to provide exterior orientation parameters which ensure the best match between the projected 3D building model and the image features. Then a tracking strategy based on the visibility prediction is introduced.

Chapter 5 introduces a concept for texture extraction and quality assessment.

Chapter 6 presents window extraction in building textures with regard to the properties of thermal data.

Chapter 7 presents the data sets used in experiments as well as the test area. The methods to assess the results are also proposed in this chapter.

Chapter 8 shows the results for the test area described in Chapter 7, which were achieved by testing and validating the methodology described in chapters 4-6. The results are structured in sections according to these chapters.

Chapter 9analyzes the results of the experiment. Problem areas and possible errors are identified. This chapter is structured in sections according to chapters 4-6.

Chapter 10 concludes the findings of the thesis and highlights the contribution of the work. This chapter also recalls the problem areas from Chapter 9 and develops ideas to overcome the weaknesses and unsolved problems found in this study.

2 State-of-the-art

Realization of the tasks outlined in Section 1.2 touches various research areas within thermography, photogrammetry, remote sensing and computer vision. State-of-the-art in those fields, which are relevant for this thesis, will be presented in the following sections. First, recent research on thermal imaging of buildings will be presented (Section 2.1), followed by a literature review on the texture mapping (Section 2.2). Next, current research is presented for sub-subjects of texture mapping that are relevant for this work, namely *model-to-image co-registration* (Section 2.3), *texture extraction* (Section 2.4) and *object detection in façade textures* (Section 2.5). Finally, the subjects and goals of this work are defined based on the literature review and identified gaps in the state-of-the-art (Section 2.6).

2.1 Thermal Imaging in Urban Environment

The work presented in this thesis belongs to the field of high resolution thermal infrared (TIR) imaging of urban environments; therefore, recent projects in this field are briefly described here.

Recent thermal investigations concentrate on human-made objects that are at risk of loosing heat. In Vienna, a test flight with the thermal sensor TABI-1800 mapping a swath by 1800 sensor elements was carried out, and 25 cm resolution data for about 30 km^2 was collected. Fig. 2.1 shows Votiv Church in Viena's city center, whereas Fig. 2.1a presents a RGB orthophoto, and Fig. 2.1b corresponding thermal orthophoto. In the presented example, a digital terrain model (DTM) was used for orthophoto creation. Hence, building façades in off-nadir view that are facing to the sensor are partially visible in the orthophoto. In this project, however, no 3D information was explicitly combined with the thermal data. The first round of collected data was applied for an investigation of heat losses in buildings, a control of the heat distribution network, and a control of heating for railroad switches [Jonst, 2013]. Heat distribution and leakage detection in district heating networks are the main objective of the work presented by Berg & Ahlberg [2014].

Some projects are dedicated solely to the thermal inspection of buildings. Fondazione Bruno Kessler conducted a project IR3D with the goal to metrically compute the heat losses of the building [Fondazione Bruno Kessler, 2014]. Another research project "Enrichment and multi-purpose visualization of building models with emphasis on thermal infrared data" was carried out at Technische Universität München (TUM). The aim of this project was to automate the acquisition of thermal data for building façades and texture extraction, as well as automatize the analysis and visualization of the thermal data in Geographic Information Systems [Meng & Stilla, 2007; Technische Universität München, 2013]. Within this project, thermal images were acquired using a TIR camera mounted on a vehicle. In the post-processing, much work has been done on relative orientation, matching with the 3D building models using a generated point cloud and automatic mosaicing of oblique image sequences in order to create high resolution thermal textures [Hoegner, 2014].
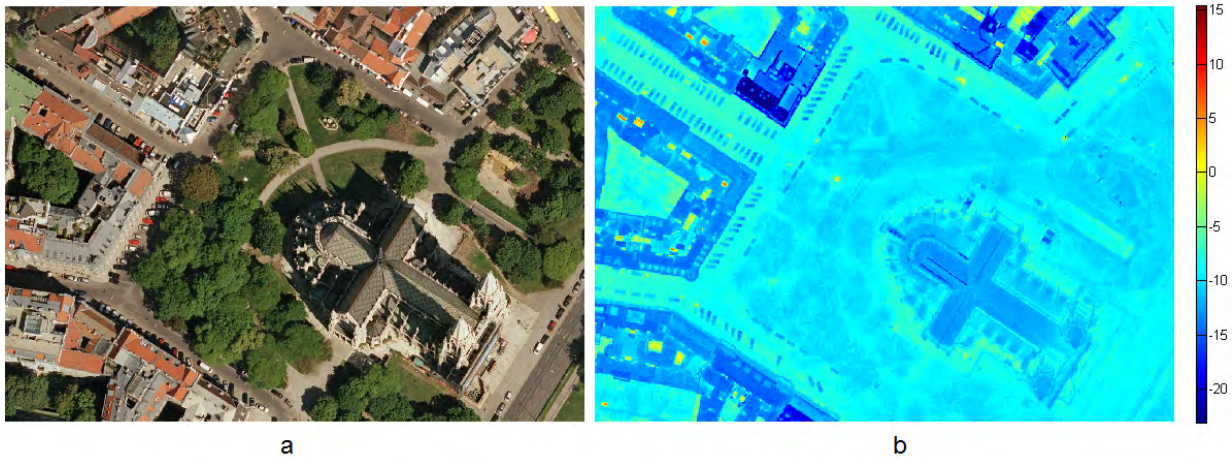
Figure 2.1: Orthophoto of Votiv Church in Vienna. a) othophoto in visible (VIS) domain; b) thermal infrared (TIR) orthophoto. Source: Stadt Wien, MA 41-Stadtvermessung `http://www.stadtvermessung.wien.at`

Researchers at Massachusetts Institute of Technology who worked on large area thermal imaging of urban areas also mounted a thermal camera on a vehicle [Chandler, 2011]. They aimed to quickly identify inefficient buildings by detecting heat losses through walls, roofs, doors and windows. Within this project, an imaging system called Automated Rapid Thermal Imaging Systems Technology (ARTIST) was developed [Phan, 2012]. Another idea presented by researchers from Jacobs University Bremen was to carry out thermal mapping in indoor environment. Their project, "ThermalMapper", aims at constructing thermal 3D models of indoor environments for building inspection [ThermalMapper, 2013]. A laser scanner and a thermal camera are mounted on a robot for mobile mapping of building interiors. Thermal information is mapped on the acquired point cloud.

Thermal data collected in urban areas can be also used for an online system with an open access. HEAT (Heat Energy Assessment Technologies), a GeoWeb service, is provided in Calgary, which can be used by house owners to view their building quality, or by maintenance companies to verify building quality and to monitoring over space and time [HEAT, 2013]. This system stores thermal images of building roofs together with address information and detected hot spots. The cost per day of heating the home and CO_2 emission are estimated based on the thermal data that was acquired with a thermal pushbroom scanner TABI-320 delivering stripes 320 pixels wide. A similar system is available for the island Jersey in the Channel Islands [States of Jersey, 2013].

Conclusion: None of the projects mentioned above has dealt with airborne oblique view thermal data and combined them with a 3D building model, which is objective of this study. This thesis is a continuation of the project "Enrichment and multi-purpose visualization of building models with emphasis on thermal infrared data" carried out at Technische Universität München in terms of extracting thermal textures for those parts of the building that cannot be seen from the street level. The first part of this study, however, focuses mainly on *co-registration of airborne thermal image sequences and 3D building models* using *linear features for matching* of those two data sets. In the second part of this study, strategies for *texture extraction and texture quality assessment are adjusted to the airborne data*, not to the terrestrial data captured in narrow street, as it was in the "Enrichment and multi-purpose visualization of building models with emphasis on thermal infrared data" project. In the third part of this work, the terrestrial textures created in

frame of that project [Hoegner, 2014] are used for window detection in order to show the difference in the difficulty compared to airborne textures.

2.2 Texture Mapping

The main objective of this study is to use texture mapping to combine TIR images and 3D building models. Texture mapping on 3D models is a widely used technique, especially in computer graphics, and results in adding an image to the existing 3D geometry. Photorealistic [Weinhaus & Devarajan, 1997; Allen et al., 2001] and non-photorealistic [Klein et al., 2000; Jahnke et al., 2008] textures, however, can be distinguished. For photorealistic texturing, the assignment of the images of a real scene to the corresponding 3D model has to be accomplished.

Texture extraction has already been implemented in several commercial software and imaging systems. One of such systems, Semi-Automated Modeling and Texturing System (SAMATS), was presented by Hegarty & Carswell [2009]. This system produces textured building models from a set of geo-referenced terrestrial images. Similarly, the system *Toposcopy* [Groneman, 2004] has been developed to create photorealistic 3D models. It uses photogrammetric methods for linking a 2D map to terrestrial images. Grenzdörffer et al. [2008] use MultiVision, a commercial software, to texture the created 3D models semi-automatically.

Textures can be extracted from the images taken in different spectral bands, e.g. visible (VIS) images [Hsu et al., 2000; Früh et al., 2004; Abdelhafiz & Niemeier, 2009; Abdelhafiz, 2009], multi-spectral images [Pelagotti et al., 2009], or infrared images [Hoegner & Stilla, 2007; Iwaszczuk et al., 2011c, 2012b]. In principle, it is also possible to combine 3D building models with Synthetic Aperture Radar (SAR) images as long as shadow and layover effects are considered [Götz, 2010].

Various imaging systems have been applied for texture mapping. Wang et al. [2008] used the system *Pictometry* for this purpose. This system consists of five cameras with one nadir-looking camera and four oblique looking cameras which are mounted on a flying platform. This system found a variety of applications including urban planing, 3D modeling, and emergency response [Karbo & Schroth, 2009]. The system PFIFF [Grenzdörffer et al., 2008] is also based on an oblique looking camera, which is integrated with a GPS receiver. Oblique view geometry require special treatment for flight planning [Grenzdörffer et al., 2008] or measurements [Höhle, 2008]. Texture mapping is also possible using a push-broom instrument. Lorenz & Döllner [2006] textured 3D building models using High Resolution Stereo Camera (HRSC) mounted on an aircraft.

Despite different properties of particular spectral bands and of various imaging systems, a generalized work-flow for the texturing process can be outlined. Fig. 2.2 shows such a work-flow, which was followed by many of the works on texture mapping on 3D building models [Hsu et al., 2000; Bornik et al., 2001; Sawhney et al., 2002; Früh et al., 2004; Song & Shan, 2004; Kada et al., 2005; Karras et al., 2007; Grenzdörffer et al., 2008; Wang et al., 2008; Hegarty & Carswell, 2009; Abdelhafiz & Niemeier, 2009] and also commercial solutions [Groneman, 2004; UVM systems, 2015]. In this figure, solid arrows represent necessary steps while dashed arrows express optional steps. The optional procedures appear as transparent boxes in this work-flow. The opaque boxes are followed by most of authors in diverse variations and with different focuses.

3D building models representing urban scenes are required to be geo-referenced for most applications; therefore, *direct geo-referencing* [Früh et al., 2004; Klinec, 2004; Kada et al., 2005; Karbo & Schroth, 2009] and/or *aerotriangulation* [Song & Shan, 2004; Grenzdörffer et al., 2008; UVM systems, 2015] are the first steps in the sketched work-flow (Fig. 2.2-1). Two main cases can then be distinguish for texture mapping and are covered by this generalized work-flow:
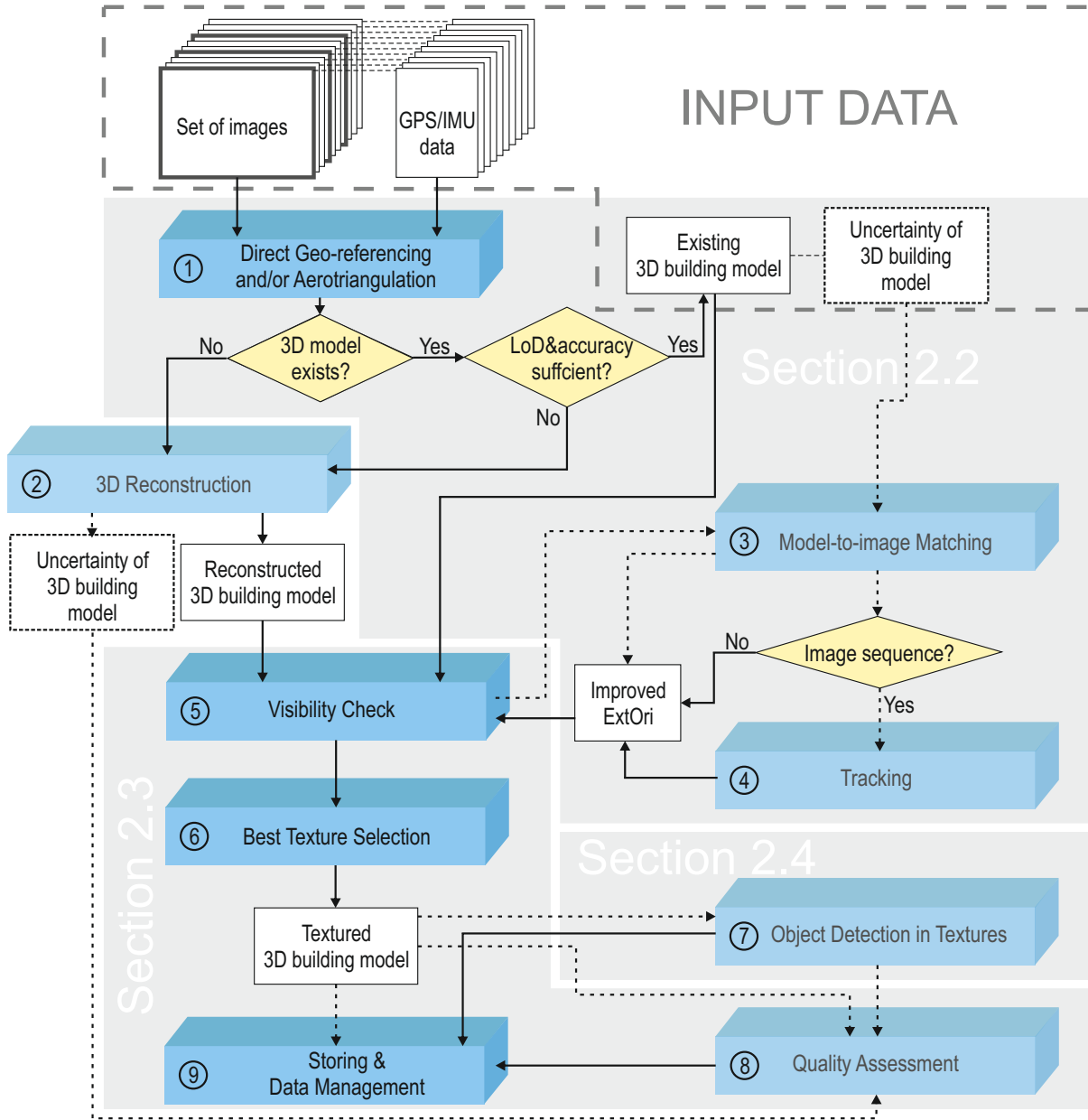
Figure 2.2: Generalized work-flow for the texturing process. The dashed line represents optional steps and the transparent boxes represent optional processes. ExtOri denotes exterior orientation of the camera. LoD denots Level of Detail.

- 3D building models are available [Hsu et al., 2000; Früh et al., 2004; Hoegner & Stilla, 2007],

- 3D building models are not available or is insufficient for the intended application [Debevec et al., 1996; Bornik et al., 2001; Kuschk, 2013].

In case of missing or insufficient 3D building models, a simultaneous *3D reconstruction* (Fig. 2.2-2) is required. Some authors proposed reconstructing of 3D building models from oblique airborne images [Lin & Nevatia, 1995; Karbo & Schroth, 2009; Panday & Gerke, 2011]. In such cases, textures for the façades can also be extracted from the same imagery. Here, the relative orientation between the images and the reconstructed 3D building model is given per se and no matching of the data sets is needed. 3D reconstruction is not the subject of this work, but it is mentioned here for the sake of completeness. Further references to this topic can be found in Kolbe [1999]; Suveg & Vosselman [2004]; Remondino & El-Hakim [2006]; Haala & Kada [2010].

In case 3D building models are available, it is required that 3D models and images are co-registered. Co-registration can be carried out by *direct geo-referencing*; however *direct geo-referencing* is often not sufficient to accurately co-register the data [Früh et al., 2004; Kada et al., 2005; Grenzdörffer et al., 2008] and is used only as approximate alignment. In airborne photogrammetry, geo-referencing can be carried out in the frame of *aerotriangulation*; however, *aerotriangulation* can require manual selection of control points and does not yield good results for stripe-wise acquired oblique images with one perspective angle [Grenzdörffer et al., 2008]. Manual selection of control points can be particularly time consuming for image sequences with a high frequency rate. Results of *aerotriangulation* for oblique thermal images carried out with four perspective angles can also be insufficient for high accuracy model-to-image co-registration [Kolecki et al., 2010]. The mismatch can be due to inaccurately estimated exterior and interior orientations of the camera or inaccuracies in the 3D building model; therefore, many authors propose *model-to-image matching* (Fig. 2.2-3) in order to improve the co-registration [Früh et al., 2004; Ding & Zakhor, 2008], which can be supported by *tracking* [Hsu et al., 2000] (Fig. 2.2-4). Automated *model-to-image matching* can also be carried out during the flight and contribute to the localization of the aircraft or vehicle.

After co-registration occurs, texture extraction can be carried out. The next necessary steps are *visibility check* (Fig. 2.2-5) and *best texture selection* (Fig. 2.2-6), which result in a textured 3D model. *Quality assessment* (Fig. 2.2-8) can also be carried out [Lorenz & Döllner, 2006; Hoegner et al., 2012]. The final step is *storing and data management* (Fig. 2.2-9), and more detailed elaborations on this topic can be found in various publications [Früh et al., 2004; Song & Shan, 2004]. Texture extraction can be also followed by *object detection* in façade textures [Hoegner & Stilla, 2009] (Fig. 2.2-7).

The presented work-flow assumes a *calibrated camera system*, which can be done in a self-calibration process or calibrated in advance using a calibration field [Faugeras et al., 1992; Hartley, 1994; Fraser, 1997; Pollefeys et al., 1998; Triggs et al., 2000; Hemayed, 2003; Remondino & Fraser, 2006; Tang et al., 2007]. All interior parameters of the camera, including lens distortions, should be known before texturing. Additionally, a lever-arm and boresight calibration [Kolecki et al., 2010; Hebel & Stilla, 2012] should be carried out. These parameters can be also determined in a self-calibration process in conjunction with exterior orientation (ExtOri) parameters [Kolecki et al., 2010].

In order to facilitate a geometrical calibration of an infrared camera, the control points have to differ from its surrounding through the emitted radiation. Special calibration fields with an electric bulb can be applied for calibration in laboratory [Luhmann et al., 2011].

Conclusion: The generalized work-flow for texture mapping presented in Fig. 2.2 is also suitable to fulfill the goals of this work and is followed in the methodology and experiments presented within this study. The focus of this thesis is, however, adjusted to the objectives of the work. It is assumed that the airborne thermal textures will be combined with other textures (terrestrial, visible) using the same 3D building model. Moreover, combination of the textures from different points in time should be possible in form of multiple textures for each face. An existing 3D building model, therefore, is used and co-registered with the images in order to achieve good fit between the data sets.

A literature review on *co-registration*, the main focus of the thesis, is found in (Section 2.3), including *direct geo-referencing* and *model-to-image- matching* also supported by *tracking*. The second focus of the thesis is *texture extraction* adapted for the requirements of thermal inspections. Thus, the literature describing the topic of texture extraction is presented in Section 2.4, and includes *visibility check, best texture selection, quality assessment storing and data management.* Since this study deals also with window detection in façade textures, *object detection in façade textures* are explored in Section 2.5.

2.3   Model-to-Image Co-registration

Model-to-image co-registration is a necessary step in texture mapping on existing building models. In this section, relevant research is presented. The studies referenced do not limit to co-registration of 3D building models with thermal images; they also include other spectral bands, particularly the visible domain. The co-registration methods that are presented, also are not restricted to 3D building models too, but the selected literature focuses heavily on urban areas and building models.

2.3.1   Direct Geo-referencing

Direct geo-referencing consists of geo-referencing using known exterior orientation (ExtOri) of the camera without taking any information from the image. Approximated exterior orientation parameters can be directly determined using global positioning system (GPS) data. In order to increase the accuracy of direct geo-referencing, an inertial navigation system (INS) is often used together with the GPS receiver. INS provides good short-term accuracy, but over time, a systematic drift occurs. Thus, the combination of GPS and INS makes it possible to avoid the INS drift and to bridge any short loss of satellite signal by GPS [Yastikli & Jacobsen, 2005]. In order to apply direct geo-referencing, a system calibration needs to be carried out. In particular the camera position and orientation are often not identical to the position and orientation registered by integrated GPS/INS, in which case the estimation of the misalignment angles (boresight parameters) and the lever arm vector is necessary [Yastikli & Jacobsen, 2005; Eugster & Nebiker, 2007; Stilla et al., 2009; Kolecki et al., 2010]. Furthermore, a transformation between coordinate systems is necessary, since most building models are stored in national coordinates, while GPS/INS navigation uses a geographic coordinate system [Legat, 2006; Skaloud & Legat, 2008]. The accuracy of georeferencing depends on the sophistication (and therefore price) of the navigation device, and on the availability of differential GPS (DGPS) corrections. Usually the accuracy of the direct geo-referencing is too low for a precise direct co-registration and texture extraction, but it can be used as approximated values to initialize further processing. Position errors in 3D building models can also be observed as a result of 3D reconstruction using inaccurately geo-referenced data and large deviations in modeling. Inaccuracies in boresight calibration can also lead to mismatches between the 3D building model and the image [Kolecki et al., 2010].

These mismatches can be reduced via a model-to-image matching, and the matching should be adaptable to the accuracy of the direct geo-referencing.

2.3.2 Model-to-Image Matching

Model-to-image matching is a widely discussed topic and various methods for implementation have been developed. Some authors [Van den Heuvel, 1998; Hu et al., 2006; Ding & Zakhor, 2008; Cham et al., 2010; Förstner, 2010b] assume that considered scenes consist of piece-wise planar surfaces with dominant directions (so called Manhattan or Legoland scenes) and calculate vanishing points of the vertical and horizontal lines. This, in combination with GPS data, can be used for the computation of exterior orientation parameters. These methods require a calibrated camera system and and the extraction of many vertical and horizontal lines in the image. Because of the lack of vertical and horizontal lines, which can be unambiguously extracted, these methods can fail in residential areas [Ding & Zakhor, 2008]. In some works [Vosselman, 1992; Eugster & Nebiker, 2009; Eugster, 2011], relational matching is applied, which considers not only the agreement between an image feature and a model feature, but also takes the relationship between features into account. These methods can also be differentiated based on the image features used for matching. For model-to-image matching in urban areas, some authors propose points [Ding & Zakhor, 2008; Wang et al., 2008; Avbelj et al., 2010], but most consider lines more natural for building structures used them for co-registration [Debevec et al., 1996; Hsu et al., 2000; Früh et al., 2004; Eugster & Nebiker, 2009]. Hybrid methods employing points and lines at the same time, however, have been also discussed [Zhang et al., 2005; Tian et al., 2008].

Some methods fit 3D models directly to the contours. Lowe [1991] proposes a method for fitting parametric 3D models to images, mainly for recognition and tracking purposes. He detected contours in the image and used them to determine projection and model parameters. This was done by minimizing the perpendicular distance from the points on the image edge to the projected model curve. Additionally, Lowe proposes a method to solve the problem when there are more unknowns than constraints on the solution based on the observations. For this purpose, he introduces prior constraints on the desired solution. Vosselman [1998] adapts the method proposed by Lowe [1991] to match parametric building primitives with aerial images in a semi-automatic way. He avoids the threshold for the gradient value by calculating the gradient perpendicular to the model edge in the surroundings of the edge and then weighting the observations with a squared grey value gradient. Panday & Gerke [2011] continue the idea to fit parametric building models using oblique aerial images. They extend parametric building models with a roof overhang and use oblique images to update building models obtained with airborne laser scanning (ALS). Nyaruhuma et al. [2012] also register 3D models with images using image gradients. They implement edge matching of 3D building models with airborne oblique images using mutual information technique for individual edges, faces, or whole buildings. In their research, they tackle the problem of identifying changes in building inventory. For this purpose, they project the 3D building model into the image and calculate the gradient direction. Then they use them to calculate the "pixel gradient direction probability density" and "edge pixel gradient direction probability density" and compute the mutual information out of these two probability densities.

Some authors believe line matching of aerial images can be done with with 2D vector data. Habbecke & Kobbelt [2010] coregister 2D cadastral maps using oblique images in three stages. In the first stage, similar to the work by Ding & Zakhor [2008], they calculate vanishing points and use them to reduce unknown parameters of exterior orientation from six to four in order to simplify the search for further parameters. They then detect lines in oblique images, which correspond to vertical lines in the object space, and assign them to map corners vertices. They use RANSAC [Fischler & Bolles, 1981] to select the correct correspondences from a larger set of

initial correspondences. Lastly, they refine the co-registration by detecting the lines horizontal in the object space and matching them across pairs of images. Using these matches, they solve bundle-adjustment-like global optimization over all camera parameters. Kawai & Saji [2007] also coregister aerial oblique images with 2D maps. First, they roughly determine the exterior orientation of the images by global registration. They support the co-registration using vertical aerial imagery at this stage. Next, they refine the co-registration locally by matching corresponding points in the oblique images and in the 2D map. They compute the projective transformation as the final result of the co-registration.

Considering different properties of objects in the infrared domain, the selection of a specific approach is often needed. Stilla et al. [2000] and Avbelj et al. [2010] propose a method for matching low resolution TIR images based on intersection points of roof edges. Roof edges are selected because they can be reliably extracted in IR images. This is related to the operation principle of the TIR cameras, which record the thermal radiation of the photographed objects. This radiation depends on objects temperature and emissivity, and the emissivity depends on the material. Building façades are usually made of materials with emissivity similar to the emissivity of the materials used for pavement and streets. Hence, the contrast between the building façades and pavements or streets is often low. For roofs, especially when made of roof tiles or sheet metal, the contrast to façades can be better used for edge detection in TIR images.

Pelagotti et al. [2009] propose a method for co-registration of 3D models with multi-spectral images using mutual information between two images. From the 3D model, a depth image is calculated and coregistered with the mutli-spectral image by maximizing the mutual information. They also deform the image to be registered using affine transformation as deformation model.

Line Matching: As mentioned above, lines are more representative for man-made objects than points; therefore, line matching is an important task for co-registering of 3D building models with images. One of the earliest examples of line based model-to-image matching was presented by Sester & Förstner [1989]. They present a concept for object location in aerial images using the orthogonal 2D sketches of buildings. The goal of this work is to determine the position of the control points in the image, such as the gable points of the roof. They interpret the 2D sketches creating 3D parametric description and use probabilistic clustering to find image lines corresponding to the lines from the sketches. They then automatically determine the exterior orientation of the aerial images. From the corresponding line pairs, they obtain the optimal solution for the exterior orientation parameters of the camera using a robust estimation. Schickler [1992] extends this approach with self-diagnosis by analyzing the final results with respect to the precision and sensitivity of the geometric configuration of the control points. Schickler [1992] also mentions the possibility of employing the RANSAC technique to find incorrectly located control point models and to predict a more likely set of matching candidates. He also presents an adaptation of this method for semi-automatic mapping of 3D objects.

In the texture mapping context, some authors also employ line based co-registration. Früh et al. [2004] propose an approach based on matching line segments with model edges. In this method, the edges are extracted in the image and the model is projected into the image from random camera positions. A rating based on line matching is calculated and the procedure is repeated for each position. The position with the highest rating is chosen for texture mapping. Many random parameter configurations have to be tested for every frame, thus the computational effort of this method is very high. No correspondence between frames is used in this approach

Lee et al. [2002] uses vanishing points to calculate the orientation of the camera and three correspondences of line segments to recover the camera position. After determining the orientation parameters, they create hypotheses about the camera position, and then they classify all image

lines and all model lines into two major directions. Finally, using the RANSAC, they calculate the position of the camera out of three correspondences of line segments. Kada et al. [2005], aiming a real time visualization of urban scenes using graphic hardware, implement a modified spatial resection with lines as described in Klinec [2004]. Klinec [2004] presents a system for pedestrian localization by model-to-image matching using lines. He uses direct linear transformation (DLT) to determine the initial camera position. He then uses modified collinearity equations with line parametrization for final solutions, as proposed by Schwermann [1995]. Cheng et al. [2013] use line matching for texture mapping in indoor environments. Assuming the planarity of the captured scene, they assign lines from image pairs using RANSAC and calculate translation between them. This translation is used for image stitching.

Hsu et al. [2000] and Sawhney et al. [2002] texture existing 3D models using a video sequence. They assume the camera pose to be known in the first frame of the sequence and predict the pose in the next frame. The correspondence between the frames is estimated using optical flow. They then search for the best camera position by minimizing the disagreement between projected edges and edges detected in the image. For this purpose, they represent the local edge strength within an image as an energy field. Next they vary the pose and try to maximize the integral of this field along the projected 3D line segment. They use the steepest descent method to find the optimal solution.

Line matching was frequently addressed in the 3D reconstruction context. Debevec et al. [1996] propose a method for reconstruction using lines marked in the image by the user. They represent lines with two vectors, the first one representing the direction of the line and the second one representing a random point on the line. This allows then to represent the projection of the line as an intersection of the plane spanned by two vectors: the vector from the point on the line showing in the line's direction and the vector from the projection center to the point on the line. They use parametric 3D primitives for reconstruction and obtain the 3D structures by minimizing the sum of the disparity between the projected edges of the primitive models and the edges marked in the image. Schenk [2004] represents lines using four parameters proposed by Roberts [1988] in 3D and defines colineraity equations for the lines using these parameters. This idea is also utilized by Meierhold et al. [2008]. Heuel & Förstner [2001] propose a methodology for the 3D reconstruction of buildings from multiple oriented images using statistical geometric reasoning for projective geometry. They group uncertain 2D and 3D entities and use them to construct further entities. Ok et al. [2012] concentrate on finding correspondences between the lines. They define a pair-wise measure for line correspondence in stereo pairs to find the initial correspondences. The final correspondences between the stereo images are established in a line-to-line matching. In this line-to-line matching, they consider a similarity measure based on a Daisy descriptor, a redundancy measure, and a pair-wise quality measure.

The majority of works assume the 3D building model to be error free. Only a few of the presented methods for model-to-image matching consider the uncertainty of 3D models [Sester & Förstner, 1989; Schickler, 1992; Lowe, 1991]. In these works, the 3D models are stored in a parameterized form; which is very useful for 3D reconstruction. These parameterized models represent simple buildings or building primitives; however, reconstructed building models are frequently modeled by polyhedra and stored in a format supporting polyhedral models, e.g. *CityGML*.

Representation of Uncertainty: Sester & Förstner [1989] and Schickler [1992] introduce uncertainty in three model parameters (width, length and slope) for a simple case of roof sketches and integrate them in the adjustment, together with the uncertainties in two parameters of 2D lines detected in the image. Luxen & Förstner [2001] present a method for optimal estimation for the projection matrix with the covariance matrix for its entries using point and line correspon-

dences. Using homogeneous coordinates, they represent 3D lines by joining of two 3D points and the projection of these lines as projection planes. In doing so, the entries of the projection matrix for points (size 3×4) is calculated, and calculation of the projection matrix for lines (size 3×6) can be avoided. Luxen & Förstner introduce the uncertainty of the 2D points and lines in the adjustment model.

Heuel & Förstner [2001] and Heuel [2002] also use a homogeneous representation of geometric uncertain entities to match line segments in order to optimally reconstruct 3D lines and group them. Heuel [2002] gives a very detailed and structured overview of the representation of uncertain entities in 2D and 3D, including points, lines, and planes and geometric reasoning. He also presents the constructions using uncertain entities and appropriate error propagation. Beder [2004, 2007] uses the same representation for grouping points and lines by statistical testing for incidence.

Meidow et al. [2009a,b] collect, evaluate, discuss, and extend various representations for uncertain geometric entities in 2D. Additionally, they provide a generic estimation procedure for multiple uncertain geometric entities with Gauss-Helmert model. They handle uncertain homogeneous vectors and their possibly singular covariance matrices by introducing constraints for the observations in addition to the conditions for the observations and parameters and restrictions for the parameters.

Förstner [2010a] introduces a minimal representation of uncertainty for points and lines in 2D and 3D using the unit sphere. A minimal representation is achieved by reducing the homogeneous coordinates by projecting them on the plane tangent to the unit sphere at the considered geometric entity. Special attention is paid to the 3D lines, which have to be reduced on a four dimensional tangent space and treated separately. He also gives an application example of the introduced representation in bundle adjustment. Förstner [2012] extends the previous work by presenting an application for statistical testing. Schneider et al. [2012] use this minimal representation of homogeneous coordinates for image and scene points in a bundle adjustment for omnidirectional and multi-view cameras. Instead of using Euclidean normalization, they integrate far points and points at infinity in the adjustment using spherical normalization. In Förstner [2010b], the minimal representation is implemented to optimally detect vanishing points.

Conclusion: In this study, co-registration will be done by improving the camera pose based on the correspondence between the model and image features. Due to the different accuracies of the direct geo-referencing, depending on the quality of navigation device, the matching should adapt its accuracy, which will be explained in this thesis. Many authors mention that linear features are most suitable for model-to-image matching in urban areas; therefore, line segments will be used for matching in this study. For this purpose, the line representation suitable for the application will be chosen. Since image features and 3D building models are expected to be uncertain, the line representation and estimation method should be selected, so that the uncertainty can be taken into consideration. This will be done by using two representations. First, the representation introduced by Schenk [2004] will be used and extended with the uncertainties of the 3D model. The estimation will be done using the Gauss-Markov model and the collinearity equations for lines in Euclidean space. Second, the representations presented in Luxen & Förstner [2001] and Heuel [2002] will be adapted for the purposes of this work, as well as the generic estimation procedure introduced by Meidow et al. [2009b].

2.4 Texture Extraction

In this work, texture extraction is understood as the process of selection and association of an image section to a face of a 3D building model, which includes visibility check and storing this selection with the 3D model. The co-registration between the 3D model and the image is assumed to be given.

Visibility check: In general, two groups of methods for visibility check can be distinguished: (i) image based methods and (ii) polygon-based methods. In image-based methods, z-buffer (depth image) or ray casting algorithms are applied. In polygon-based (vector-based) approaches, polygon clipping is used. Typically, all polygons (triangles) are projected into the image plane and are intersected [Kuzmin et al., 2004; Hanusch, 2008]. The depth-buffer method is a basic method for removing hidden surfaces adopted from computer graphics. The depth-buffer is created by projecting a polygon of the 3D scene into the image and storing that pixels are occupied by the polygon together with the distance to the polygon from each pixel. This procedure is repeated with the other polygons and the occupancy of the pixels is overwritten when the next projected polygon has a lower distance to the image at this pixel. This method is revisited often with some variations [Früh et al., 2004; Karras et al., 2007]. In ray casting technique, the depth image is generated by tracing the rays sent from the projection center through each pixel of the image plane. Those rays are intersected with all surfaces of the 3D scene, and the intersections are sorted due to the distance to the image plane. The surface is marked as visible at a certain pixel if its intersection with the corresponding ray was closest to the image plane [Hoegner et al., 2007]. Bénitez & Baillard [2009] introduce and compare three methods: 2D ray tracing, 3D z-buffering, and 3D ray tracing. The 2D method is not useful for the cases when a higher building is visible behind a lower one. Vallet & Houzay [2011] present a method to efficiently compute the visibility using GPU. Abdelhafiz & Niemeier [2009] and Abdelhafiz [2009] integrate digital images and laser scanning point clouds. They use a *Multi Layer 3DImage (ML3DImage)* algorithm, which classifies the visibility on two stages: point stage and surface stage. Using model-based visibility check, only self-occlusions can be detected. For high quality texture mapping, especially when certain objects, such as windows or heat leakages need to be detected in the texture, non-predictable (un-modeled) occlusions have to be taken into account. For this purpose, multiple images [Böhm, 2004; Abdelhafiz & Niemeier, 2009; Abdelhafiz, 2009] or laser point clouds [Bénitez et al., 2010] are utilized.

Best Texture Selection: Best texture selection is an important step of texture extraction because it enables texturing with high quality textures. This becomes particularly important for texturing using an image sequence. In literature, different strategies for the best texture selection can be distinguished. Some authors propose using only one parameter as selection criteria. Bénitez & Baillard [2009] selected textures based on visibility only; however, they mention the need to extend their method with quality criteria. Debevec et al. [1996] introduced *View-Dependent Texture-Mapping*. In this approach, the angle between the viewing direction of the camera and the normal of the face were calculated. For texture mapping, the image with the smallest angle between the viewing direction and the normal were selected. Similarly Wang et al. [2008] use the viewing angle to select the oblique image for a façade texture. In some studies more then one parameter is used to calculate the quality measure. Früh et al. [2004] defined the quality measure based on resolution, occlusion, viewing angle and coherence with the neighboring triangle. Some authors discussed the problem of radiometric differences between two frames and solve it by appropriate *texture blending*. Texture blending combines multiple images in order to achieve a high quality texture for one face. Früh et al. [2004] implemented texture blending by favoring neighboring triangles of the same model face to be extracted from the same image.

Karras et al. [2007] introduced a weighted texture blending. Their method regulates the local radiometric contribution of each image. Outlying color values are rejected automatically using statistical tests. Lorenz [2011] proposed a texel-wise selection and blending, taking into account the localization in the oblique view image, occlusion, and viewing angle for each texel separately.

Quality Assessment of Building Textures: In most studies, the texture quality was introduced as a value used to select the best texture. Such quality calculated for the selection procedure can be stored with the texture. It is an abstract value, however, which can be interpreted to compare the quality between faces, but does not give information any about the level of detail of the texture and its fit to the 3D model. Some researchers, therefore, calculate local resolution for every pixel. Lorenz & Döllner [2006] analyzed the quality of texture extracted from airborne images taken with an HRSC camera and created quality maps consisting of local effective resolution. Similar resolution maps for textures are also presented in Hoegner & Stilla [2007]. Hoegner et al. [2012] assess the matching quality between the image sequence and the building model by analyzing the extracted textures. Textures from different sequences at different times and with different orientation parameters are compared through correlation and assessed visually. This method does not give any independent measure that could express the quality of fit between the model and the extracted texture.

Storing and Data Management: Another issue in texture mapping is the storage of extracted textures. Some authors [Früh et al., 2004; Kaul & Bohn, 2008] use a texture atlas (image containing all textures) to avoid storing many small images independently. Storing textures in a database and integrating them with a web service is also conceivable. Eugster & Nebiker [2008] integrate the textured model within the Virtual Globe. In the last decade, the standard *CityGML* for storage of 3D building models has been developed [Gröger & Plümer, 2012]. In this standard, the faces do not have to be triangulated. It is possible to store one face as a polygon or even a set of polygons. *CityGML* makes it possible to associate multiple textures with one face by assigning them to different themes (e.g. spring or winter), which can be used to store thermal textures captured in different points in time.

Conclusion: In the study *CityGML* will be used because of its ability to store multiple textures. The majority of the methods for visibility check presented above, however, are based on triangulated models and do not deal with polygon representation, which is allowed for example in *CityGML* [Gröger & Plümer, 2012]. Such polygon representation is also more useful for further interpreting the extracted textures. The described methods do not take into account the permanently occluded faces or their parts, which are typical for densely build-up areas, and they are developed for VIS images - that is, for good user perception and not for TIR images, where each pixel contain a radiation measurement. Strategies to overcome these difficulties will be developed in this thesis.

2.5   Object Detection in Façade Textures

Due to the objectives of this thesis, in this section, only the small area of object detection will be presented, namely the detection of objects in façade textures. Since the goal is to detect the windows in TIR building textures, the works on TIR images will first be presented. Literature about visible image findings will be also reviewed.

Klingert [2005] works with terrestrial TIR façade images and aims to detect heat leakage using basic image processing operations, namely pixel-oriented image segmentation with thresholding. Hoegner & Stilla [2009] analyze rectified thermal textures and try to detect thermal leakages by

detecting edges and using a region growing approach. Because the windows reflect the surroundings, in some camera configurations whole windows or their parts can appear similar to thermal leakages. In order to avoid confusion, the window areas are masked by combining the forward and backward view textures. This method is effective since windows lay on the plane slightly behind the façade plane. Sirmacek et al. [2011] present an approach to detect windows and doors, as well as thermal leakages. They use L-shapes detected in thermal textures. L-shapes are grouped to build rectangular structures. According to the defined minimal and maximal object size, windows are either accepted or rejected. The heat leakages are detected similarly to Hoegner & Stilla [2009] using a combination of morphological operators and reject leakages that lay inside the previously detected windows and doors.

Much more studies on façade image interpretation have been presented in visible domain. For example, a joint research project called *eTRIMS* (E-Training for Interpreting Images of Man-Made Scenes) which studies façade classification, has been established in Europe [Förstner & Korč, 2009; Förstner et al., 2009]. The focus of this project is to automatically recognize the main elements of a man-made urban scene, such as street, vegetation, windows, and doors. Classification methods using graphical models [Yang, 2011; Drauschke, 2011], boosting classifiers [Grabner et al., 2008] or support vector machines [Bochko & Petrou, 2007] are applied for this purpose. Others concentrate more on façade structure. Čech & Šára [2008, 2009] classified window panes by Maximum *A-posteriori* Probability Labeling, forcing them to be rectangles. Wenzel et al. [2007, 2008] search for symmetries in the rectified image and clusters detected features to interpret the façade.

Ripperda & Brenner [2006]; Ripperda [2008, 2010] used grammars to describe the façades. Their data-driven method is used to derive the distribution of façade attributes (e.g. the position of windows) and use these distributions to create rules of façade appearance. In order to construct these rules, they proposed reversible jump Markov chain Monte Carlo (rjMCMC) [Green, 1995]. MCMC based methods were also introduced by Dick et al. [2004] and Mayer & Reznik [2007]. Reznik & Mayer [2007] used implicit shape models introduced by Leibe & Schiele [2004] to detect and delineate windows using alarge set of training data. Mayer & Reznik [2008] extend this methodology with a self diagnostic algorithm. Werner & Zisserman [2002] use regular structure primitives like vanishing points or symmetry detected in uncalibrated images.

Meixner & Leberl [2010, 2011] demonstrate the potential of nadir view airborne images for façade interpretation and argue that airborne oblique images are more suitable for this purpose. However, façades are well visible in nadir view images only if they are in off-nadir areas, namely in the marginal area of the image.

Some authors aim at building façade reconstruction from 3D point clouds. Becker [2009] and Becker & Haala [2009] propose quality dependent façade reconstruction from LiDAR point clouds based on a formal grammar. They combine bottom-up and top-down approaches and integrate the rules into the data-driven reconstruction. Tuttas & Stilla [2011] use multi-aspect airborne laser scanning data in forward looking view. The points lying behind the façade plane are taken for façade reconstruction, and Fourier transformation is applied to find repetitive structures.

Parallel to the work presented in this dissertation, Schack et al. [2012] developed a method based on lattice detection and applied it to persistent scatterer point clouds and oblique aerial imagery. At the same time, Wenzel & Förstner [2012] present a method for interpreting the façade images by aggregating low-level features, such as line segments. They select aggregations significant for object detection and use them for learning and classification.

Conclusion: Façade image interpretation has been intensively discussed over the last decade and many good performing methods have been presented; however, most of the tests have been

done using VIS data. Besides, the majority of those methods needs training data, which is not available for TIR images at this moment. The few publications presenting results in TIR data were concentrating on high resolution terrestrial images. No experiments were performed using airborne TIR textures.

2.6   Subjects and Goals of the Work

Summing up the state-of-the-art presented in Sections 2.1 - 2.5, lot of work has been done on model-to-image matching, texture extraction, and interpretation of the building textures. However, still some unsolved problems remain.

The majority of the solutions for model-to-image matching do not take the errors and uncertainties of the 3D models into account. Usually, textures are extracted for triangulated models and existing methods do not consider boundary representation based on polygons. Moreover, only few authors attempt to assess the quality of extracted textures. Most of the methods for façade reconstruction are based on classification or learning and require training data. Finally, almost all the methods are optimized for the visible domain, and do not investigate other spectral bands. Accordingly, this thesis follows three main goals.

First, it develops a methodology for model-to-image matching which take errors and uncertainties of the 3D building model into account. Line segments are utilized for the matching, as they are the most suitable features for this purpose. However, their representation with uncertainties is more challenging than for points. To achieve the best fit, optimal exterior parameters of a calibrated camera are determined. Additionally, it utilize the properties of image sequences and investigates the benefit of them.

Second, it shows a texture mapping strategy suitable for thermal building inspections and allows for assessment of accuracy and quality of the extracted textures. It uses *a-priori* knowledge from the 3D building model to determine the visibility and quality of the data.

Third, it presents an interpretation method for thermal textures with emphasis on window detection based on lattice reconstruction. Mainly, the problem of blurred edges, low contrast and low resolution of thermal data is focused.

Since the methodology presented in this thesis is developed to extract textures for thermal inspection of building, it shows experiments on thermal infrared data set and discusses the results in this context, taking into account the suitability and quality of the achieved results.

Specifically, the following research questions will be investigated:

- Will co-registration results significantly improve if uncertainties of 3D models and of image features are considered in the estimation?

- To what extent does line tracking support model-to-image matching?

- What is the potential for the textures extracted from airborne thermal images to be used for window detection?

3 Theoretical Background

In this chapter, the theoretical background for Thermal Imaging (Section 3.1), Photogrammetry (Section 3.2), Image Processing (Section 3.3) and Statistics (Section 3.4) are presented. The material presented in this chapter is used in Chapters 4-6 as a basis for the developed methods presented in this study.

3.1 Thermal Imaging

This section should help the reader to understand the principle and usefulness of thermography in various areas of science and engineering. First, the scope of application is presented (Section 3.1.1). Then, the thermography principle is briefly explained (Section 3.1.2). Finally, the influence of the environment during the temperature measurement is described (Section 3.1.3.

3.1.1 Applications of Thermal Images

Thermal imaging is widely used in various fields of science and engineering. It finds application in military, medicine [Bronzino & Diakides, 2008], chemistry [Vollmer & Möllmann, 2010] and civil engineering [Feldmeier & Rossa, 2009; Fouad & Richter, 2012]. When mounted on airborne and space-borne flying platforms, thermal sensors can also be used for city climate analysis with heat islands detection [Weng, 2009] or for detecting moving objects and traffic monitoring [Stilla & Michaelsen, 2002; Hinz, 2004; Kirchhof & Stilla, 2006]. Another important application of thermal images is thermography of buildings [Weber, 1982].

Fig. 3.1 shows an exemplary aerial image of an urban area (Fig. 3.1a) and a TIR image of the same scene (Fig. 3.1b). In the TIR image, small warm objects such as a chimney (Fig. 3.1c) and street lighting (Fig. 3.1d) can be recognized as spots with higher intensity than their surroundings. Static, parked cars appear as dark spots. Static light spots in a row of dark spots suggest one of the cars had been recently driven (Fig. 3.1e-f). A light spot on the road surface indicates a car had recently driven away (Fig. 3.1g).

3.1.2 Thermographic Definitions

Several books and publications describe the principles, potentials, and challenges of thermography [Wallrabe, 2001; Gaussorgues, 1994; Minkina & Dudzik, 2009]. Thermal imaging is based on the fact that every object with a temperature greater than absolute zero emits thermal radiation as electromagnetic waves. This electromagnetic radiation is by nature similar to visible light, radio waves, X-radiation, and gamma-radiation. The difference between these radiations is their wavelength or frequency (Fig.3.2).

Thermal radiation is invisible to humans, but it can be captured by cameras working in the infrared spectrum. Infrared radiation is defined as radiation with wavelength λ=0.78-1000 [μm] (Fig. 3.2) and is classified into three spectral ranges: IR-A (λ=0.78-3 [μm]), IR-B (λ=3-5 [μm])
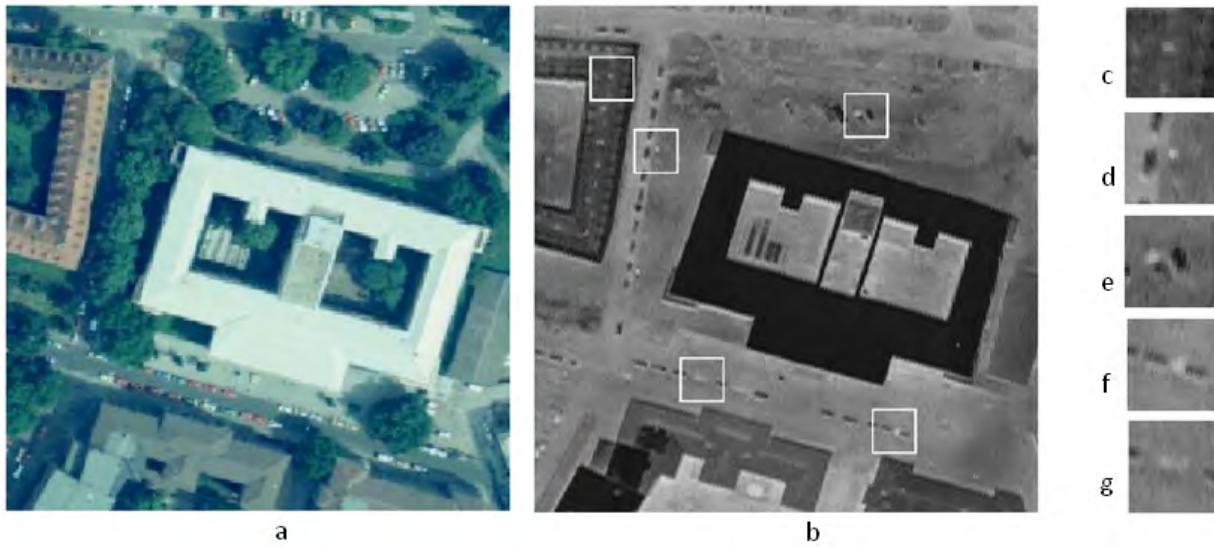
Figure 3.1: Visible light (VIS) and thermal infrared (TIR) aerial image of an urban area: a) VIS aerial image; b) TIR image of the same area; c) chimney; d) street lighting; e)-f) parked cars; g) warm spot caused by a car that has just left (source: Stilla & Michaelsen [2002])
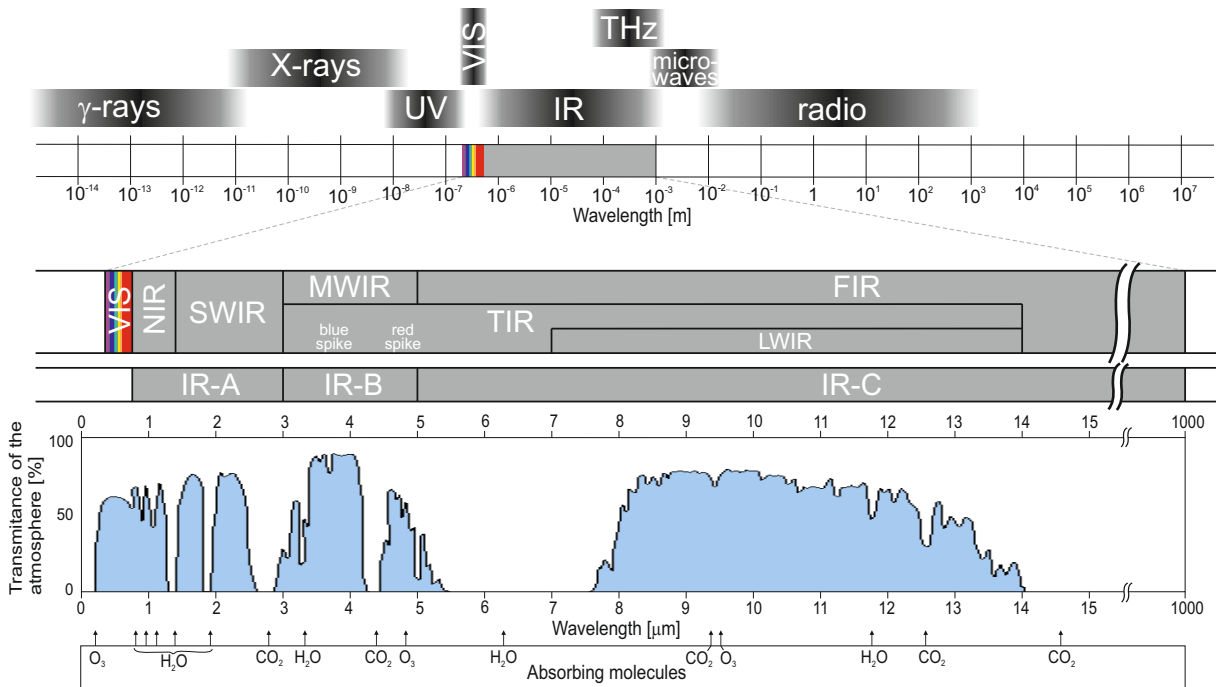
Figure 3.2: Infrared radiation: Infrared radiation is a part of the electromagnetic spectrum with wavelength $\lambda=0.78$-1000 [μm]. For thermal imaging, so called thermal infrared (TIR $\lambda=3$-14 [μm]) is used. However, only part of the TIR radiation can be observed through the atmosphere, as some bands are absorbed by molecules (water, carbon dioxide, ozone) contained in the air. The blue colored graph presents an approximate transmission of the atmosphere of Earth.(The transmission graph after Wikipedia [2015])

and IR-C (λ=5-1000 [μm]), as defined in DIN 5031. In literature, other classification are also found, so near infrared (NIR) for λ=0.75-1.4 [μm], short-wavelength infrared (SWIR) for λ=1.4-3 [μm], mid-wavelength infrared (MWIR) for λ=3-5 [μm] and far infrared (FIR) for λ=5-1000 [μm] can be distinguished. For thermal imaging, the spectral range with λ=3-14 [μm] is particularly important and is often called thermal infrared (TIR), whereas the spectral range with λ=7-14 [μm] is called long wave infrared (LWIR).

Thermal imaging is possible by sensor elements that measure radiation using different physical effects [Wallrabe, 2001], representing it as digital numbers and transforming it into grey value or color representation. In such image, it is possible to distinguish different objects because of the differences of temperature or emissivity between the objects in the captured scene. This enables intensity graduation and visualization. The ability of a camera to distinguish differences in emitted radiations and represent them as intensities is called radiometric resolution. Nowadays, thermal cameras are able to depict differences in temperature up to 0.01 [K].

The radiation measured by such cameras is usually a composition of emission, reflection, and transmission. Accordingly, each object can be characterized by three factors: emissivity component, reflection component and transmission component. The sum of these three components is equal to 1. The emissivity component is equal to the spectral absorption component. Black bodies have the emissivity component equal to 1, which means 100% of the captured radiation is caused by the temperature. No reflected or transmitted radiation reaches the camera. All other matters are characterized by all three components. Thermal radiation can also be described by radiant energy Q and by radiant flux $\Phi = Q/t$, where t is time. On the way between the imaged object and the thermal sensor, the atmosphere attenuates the radiant flux. In a homogeneous and isotropic atmosphere, the reduction of radiant flux is specified by an exponential function along the distance d. The reduction of the radiant flux depends on the wavelength (Fig.3.2). The reduction of the flux depends on weather and climate and is specific for a particular wavelength. Atmosphere absorption is mainly caused by CO_2 molecules, steam and diffusion at particles such as water droplets and fine dust.

Radiation in some of the spectral bands is completely absorbed by the Earth's atmosphere. The intervals of wavelengths in which the radiation is transmitted through the atmosphere are called the *atmospheric window*. As shown in Fig. 3.2, the atmospheric window for infrared radiation (so called infrared window) includes two bands with λ=0.78-5 [μm] and λ=8-14 [μm], whereas the spectral band with λ=3-5 [μm] and λ=8-14 [μm] belong to thermal infrared. It can be noticed that the band with λ=3-5 [μm] is split by small gap at 4.2 [μm] into the *blue spike* and *red spike*. Fig.3.2 demonstrates how the atmosphere absorbs fewer radiation in *blue spike*, thus imaging from a far distance is more advantageous at this band than in *red spike* or in LWIR. However, on shorter distances (< 2 [km]), images captured in *red spike* or in LWIR have good contrast. The solar radiation has a very small impact at LWIR. *Red spike* MWIR also has similar properties. Some sun reflections can still be observed also in *red spike* and LWIR. The sun warms up the buildings and the measured temperature is a mixture of these effects. Thermal inspection of buildings should be conducted between sunset and sunrise. Typically, such inspections are undertaken when the temperature outside is in the range of a few degrees so that the heating in buildings is already at work. However, at the time of inspection, no snow should lie close to the buildings because it can distort the measurements.

3.1.3 Temperature Measurement

The exact determination of surface temperature is difficult because of the complex influence of atmospheric conditions [Meier et al., 2011], especially due to the diffusion of the radiation by dust particles. However, for thermal inspections of buildings, knowledge about the exact temperature

of the building hull is not necessary. Thermal leakages generate temperature differences within a wall. These leakages, therefore, are detected in the thermal image as bright warm spots on an otherwise cold background. Detection of thermal leakages can be carried out manually or image processing algorithms can be used; nevertheless, strongly and specularly (mirror-like) reflecting surfaces such as glass or metal, lead to problems in thermal inspections. When photographing such objects, the temperature of the sun, of the sky, or of other objects is usually captured. Specularly reflective materials, such as glass or aluminum, are widely used in buildings, particularly for windows. It is crucial, therefore, to first detect windows and then detect heat leakages in building façades so that these areas will be excluded in subsequent leakage detection tests.

3.2   Photogrammetric Definitions

Central projection in space is essential for photogrammetry [McGlone et al., 2004; Luhmann et al., 2006; Kraus, 2007] and also for computer vision [Hartley & Zisserman, 2004]. Both disciplines use the pinhole camera model, but often there are differences in the definitions of the coordinate systems used by those two communities. In general, we can say that in a pinhole-camera model, an object point in the 3D space $\boldsymbol{X} = (X, Y, Z)$ is mapped on the image plane as $\boldsymbol{x'} = (x', y')$.

In photogrammetry, the projection is typically expressed using collinearity equations. Given the principal distance (camera constant) c_k, the principle point $(x_0, y_0)^T$, a rotation matrix $R = (r_{ij})$ and a projection center $(X_0, Y_0, Z_0)^T$, the image coordinates (x', y') of an object point $(X, Y, Z)^T$ can be written as

$$x' - x_0 = -c_k \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta x', \qquad (3.1)$$

$$y' - y_0 = -c_k \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta y', \qquad (3.2)$$

where $\Delta x'$ and $\Delta y'$ are the lens distortions in x and y direction respectively. Parameters c_k, x_0, y_0, $\Delta x'$ and $\Delta y'$ are called interior orientation parameters, and parameters $X_0, Y_0, Z_0, \omega, \phi$ and κ are called exterior orientation parameters, whereas R is a function of ω, ϕ and κ and R is rotating a point in the camera coordinate system to the object coordinate system

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + m \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x' - x_0 - \Delta x' \\ y' - y_0 - \Delta y' \\ -c_k \end{bmatrix}, \qquad (3.3)$$

where m is a scale parameter. Hence, the inverse R^{-1} is used for rotation from the object coordinate system to the camera coordinate system [Luhmann et al., 2006]. Here the image coordinate system is defined as a 3D Euclidean coordinates system with the origin in the projection center $(X_0, Y_0, Z_0)^T$, Z-axis perpendicular to the image plane and pointing away from the photographed scene. The viewing direction, therefore, can be defined as $-Z$.

In computer vision, the camera coordinate system is usually defined similarly to the camera coordinate system used in photogrammetry. It is a 3D Euclidean coordinate system with the origin in the projection center $\boldsymbol{O}$ and the viewing direction perpendicular to the image plane. Here, however, the Z-axis is pointing in the viewing direction [Hartley & Zisserman, 2004]. In computer vision, the rotation matrix R is typically rotating an object point $\boldsymbol{X}$ to the camera coordinate system. The mapping of the object point $\boldsymbol{X}$ on a plane is done by rotation R and translation to $\boldsymbol{O}$ using

$$\boldsymbol{X}_{cam} = R(\boldsymbol{X} - \boldsymbol{O}). \qquad (3.4)$$

Then, $\boldsymbol{X}_{cam}$ is projected into the image using homogeneous coordinates

$$\mathbf{x} = \mathsf{K} \begin{bmatrix} I_3 & 0_{3,1} \end{bmatrix} \mathbf{X}_{cam} = \mathsf{K} R \begin{bmatrix} I_3 & -\boldsymbol{O} \end{bmatrix} \mathbf{X}, \tag{3.5}$$

where $\mathbf{X} = \begin{bmatrix} \boldsymbol{X}^T & 1 \end{bmatrix}^T$ is the object point $\boldsymbol{X}$ in the object coordinate system represented in homogeneous coordinates and $\mathbf{X}_{cam} = \begin{bmatrix} \boldsymbol{X}_{cam}^T & 1 \end{bmatrix}^T$ is the same point in the camera coordinate system, also represented in the homogeneous coordinates. The camera matrix K is defined as

$$\mathsf{K} = \begin{bmatrix} c_k & s & x_0 \\ 0 & c_k & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.6}$$

where s is the skew parameter, which is equal to zero for most cameras [Hartley & Zisserman, 2004]. The projection matrix is defined as $\mathsf{P} = \mathsf{K} R \begin{bmatrix} I_3 & -\boldsymbol{O} \end{bmatrix}$, then

$$\mathbf{x} = \mathsf{P}\mathbf{X}. \tag{3.7}$$

This way of formulating central projection leads to a linear equation system, which is easier to deal with than the collinearity equations (non-linear system). Instead, the collinearity equations enable us to model more camera parameters related to the lens distortions.

3.3 Image Processing Definitions

Digital image processing provides many meaningful methods for the automation of photogrammetric and remote sensing tasks [Haberäcker, 1987; Blackledge, 1997; Jensen, 2004; Gonzalez & Woods, 2008; Parker, 2011]. In this section, only the approaches used in this thesis are briefly presented.

3.3.1 Cross-correlation and Masked Correlation

Cross-correlation is widely used in image processing and in image analysis to express the similarity of two image sections [Luhmann et al., 2006]. Cross-correlations coefficient of a template $\boldsymbol{t}$ and image section $\boldsymbol{g}$ is given as

$$c = \frac{1}{n} \sum_i \frac{(g_i - \bar{g})(t_i - \bar{t})}{\sigma_g \sigma_t}, \tag{3.8}$$

where n is the number of pixels in the template t, $\bar{g}$ is the average of $\boldsymbol{g}$, $\bar{t}$ is the average of $\boldsymbol{t}$, g_i is i^{th} element of $\boldsymbol{g}$, t_i is i^{th} element of $\boldsymbol{t}$,

$$\sigma_g = \sqrt{\frac{\sum_i (g_i - \bar{g})^2}{n}} \tag{3.9}$$

and

$$\sigma_t = \sqrt{\frac{\sum_i (t_i - \bar{t})^2}{n}}. \tag{3.10}$$

If the template for the correlation is a binary mask, then (3.8) can be modified, as presented in Stilla [1993], to

$$c = sgn(\rho_\oplus - \rho_\ominus) sgn(\overline{g_\oplus} - \overline{g_\ominus}) \frac{1}{\sqrt{\frac{m}{m_\ominus} \left(\frac{\sigma_\oplus}{\overline{g_\oplus} - \overline{g_\ominus}}\right)^2 + \frac{m}{m_\oplus} \left(\frac{\sigma_\ominus}{\overline{g_\oplus} - \overline{g_\ominus}}\right)^2 + 1}}, \tag{3.11}$$

where $\rho_\oplus$ denotes value of *on* mask, $\rho_\ominus$ denotes value of *off* mask, $\overline{g_\oplus}$ denotes mean value of intensity values in the image covered by *on* mask, $\overline{g_\ominus}$ denotes mean value of intensity values in the image covered by *off* mask, $m_\oplus$ denotes number of *on* pixels in the mask, $m_\ominus$ denotes number of *off* pixels in the mask, m denotes number of *on* and *off* pixels in the mask, $\sigma_\oplus$ denotes standard deviation of intensity values covered by the *on* mask and $\sigma_\ominus$ denotes standard deviation of intensity values covered by the *off* mask. This formula enables faster calculation in case of a binary template.

3.3.2   Edge Detectors

Edge detectors belong to core methods in image processing. Apart from many available algorithms, the techniques based on the first-order derivative are used in many applications. Typically, in such techniques, the *strength* of the edge is expressed using the gradient magnitude which is calculated as

$$g = \sqrt{g_x^2 + g_y^2}, \tag{3.12}$$

where g_x^2 denotes first derivative in the horizontal direction and g_y^2 denotes first derivative in the vertical direction. g_x^2 and g_y^2 are usually calculated using edge detection operator e.g., Sobel operator. The direction of the gradient is calculated as

$$\phi = \text{atan2}\left(g_x, g_y\right). \tag{3.13}$$

In practice, the Gaussian blur is often used as a pre-processing step for edge detection. This allows reducing the noise in the image and reduces multiple false detections.

3.3.3   Region Segmentation with Dynamic Threshold based on Local Image Parameters

Image segmentation can be divided into pixel-, region- and model-based methods. For pre-processing, the pixel-oriented methods, such as simple, histogram based, dynamic and hysteresis thresholding are commonly used. The principle of dynamic threshold is based on the assumption that the searched object has a homogeneous intensity which differs from the local surroundings[Haberäcker, 1987; Gonzalez & Woods, 2008].

In this approach, the background image $\mathcal{B} : \mathbf{B}$ is calculated by smoothing the original image $I : \mathbf{I}$ with a mask $\mathcal{G} : \mathbf{G}$. The size of $\mathbf{G}$ is defined to be approximately double the size of expected object. $\mathbf{G}$ is often a mean mask [Haberäcker, 1987] or a mask that is a linear combination of the mean and the standard deviation [Gonzalez & Woods, 2008].

Segmentation is carried out based on the difference per pixel between the original image $\mathbf{I}$ and the background image $\mathbf{B}$ as follows

$$\mathbf{S}(r,c) = \begin{cases} 1, & \text{if } \mathbf{I}(r,c) - \mathbf{B}(r,c) \leq t \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

for segmentation of light objects and

$$\mathbf{S}(r,c) = \begin{cases} 1, & \text{if } \mathbf{B}(r,c) - \mathbf{I}(r,c) \leq t \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

for segmentation of dark objects, where $\mathbf{S}$ is the binary result image, (r,c) is the row and column index and t is the threshold. The threshold t expresses the minimal difference between the object and the background. Light objects are objects with high intensity values, while dark objects have low intensity values.

3.4 Statistical Definitions

In this section, selected topics of estimation theory are briefly sketched to provide some theoretical background knowledge necessary to understand the content of this study, including some topics of estimation theory. Specifically, least square adjustment and robust estimators (RANSAC and clustering), representation of uncertainty of geometric entities, and statistical testing are presented.

3.4.1 Estimation Theory

Estimation theory provides a methodology for estimating unknown parameters from a given set of observations [Koch, 1999; McGlone et al., 2004; Förstner & Wrobel, 2016]. Among others, the least square estimation is commonly used to fit unknown parameters of a mathematical model to the data by minimizing a cost function:

$$\boldsymbol{\Omega} = (\mathbf{y} - s[E(\mathbf{y})])^\mathsf{T}(\mathbf{y} - s[E(\mathbf{y})]) \to \min, \tag{3.16}$$

where $\mathbf{y}$ is the observation vector and $s[E(\mathbf{y})]]$ is the function of observations (estimator) [Koch, 1999]. In addition to the mathematical model, a stochastic model can be introduced in the least square adjustment by minimizing

$$\boldsymbol{\Omega} = (\mathbf{y} - s[E(\mathbf{y})])^\mathsf{T} Q_{\mathbf{yy}}^{-1}(\mathbf{y} - s[E(\mathbf{y})]) \to \min, \tag{3.17}$$

where $Q_{\mathbf{yy}} = \boldsymbol{\Sigma}_{\mathbf{yy}}/\sigma_0^2$ is the weight coefficient matrix and $\boldsymbol{\Sigma}_{\mathbf{yy}}$ is the covariance matrix of the observations. This makes it possible to take the uncertainty of the data into consideration. Typically, for the least square estimation, one of two regression models is used: the Gauss-Markov Model or the Gauss-Helmert Model. The following explanation of these two approaches is based on [Koch, 1999; Niemeier, 2008]

Gauss-Markov Model

Assume A is a matrix of given coefficients of a linear model. The Gauss-Markov model is formulated as

$$A\boldsymbol{\beta} = E(\mathbf{y}), \tag{3.18}$$

where $\boldsymbol{\beta}$ is the vector of unknown parameters. Introducing the random vector $\boldsymbol{v}$ of the errors to $\mathbf{y}$ with $E(\boldsymbol{v}) = 0$, the models is represented by

$$A\boldsymbol{\beta} = \mathbf{y} + \boldsymbol{v}. \tag{3.19}$$

In order to find the optimal solution for $\boldsymbol{\beta}$, the cost function

$$\boldsymbol{\Omega} = \mathbf{y}^\mathsf{T} Q_{\mathbf{yy}}^{-1} \boldsymbol{v} = (\mathbf{y} - A\boldsymbol{\beta})^\mathsf{T} Q_{\mathbf{yy}}^{-1}(\mathbf{y} - A\boldsymbol{\beta}) \to \min, \tag{3.20}$$

has to be minimized using

$$\widehat{\boldsymbol{\beta}} = (A^\mathsf{T} Q_{\mathbf{yy}}^{-1} A)^{-1} A Q_{\mathbf{yy}}^{-1} \mathbf{y}. \tag{3.21}$$

The covariance matrix for estimated $\widehat{\boldsymbol{\beta}}$ is calculated as

$$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}} = \sigma_0^2 (A^\mathsf{T} Q_{\mathbf{yy}}^{-1} A)^{-1}. \tag{3.22}$$

Gauss-Helmert Model

The linear Gauss-Helmert model is represented by a set of functions $\mathbf{g_i}(\boldsymbol{\beta}, \mathbf{y})$:

$$\mathbf{g_i}(\boldsymbol{\beta}, \mathbf{y}) = A\boldsymbol{\beta} + B\mathbf{y} = 0. \tag{3.23}$$

The optimal solution is found by minimizing the Lagrange function

$$\mathcal{L} = \boldsymbol{v}^\mathsf{T} Q_{\mathbf{yy}}^{-1} \boldsymbol{v} - 2 \sum_i \boldsymbol{k_i} \mathbf{g_i}(\boldsymbol{\beta}, \mathbf{y}). \tag{3.24}$$

For non-linear problems, both the Gauss-Markov and the Gauss-Helmert models have to be linearized using first derivatives locally at the mean. The solution has to be calculated iteratively and the approximated values for unknown parameters are needed.

RANSAC

RANdom SAmple Consensus - RANSAC [Fischler & Bolles, 1981] is a robust estimation approach that fits parameters of a mathematical model using a random selection of observations. It is assumed that the data includes observations that fulfill a mathematical model (inliers) and observations that do not fit this model (outliers); however, the inliers can be subject to noise. In this approach, a solution is calculated based on minimal number of observations, which are treated as hypothetical inliers. It is tested among all observations to determine how many fit with the hypothetical solution. This procedure is repeated k-times and the solution with highest number of inliers is selected. This makes it possible to achieve a high likelihood of hitting the correct set of observations. The number of repetition k for a given probability p that RANSAC makes at least one error free selection is calculated as

$$k = \frac{\log(1-p)}{\log(1-(1-\epsilon)^r)}, \tag{3.25}$$

where $\epsilon \in (0,1)$ is the outlier rate and r is the minimal number of observation necessary to solve the task.

Clustering

Clustering is a robust estimation method based on filling an accumulator representing the parameter space [Förstner & Wrobel, 2016]. The dimension of the accumulator is equal to the number of the unknown parameters. Each observation votes for one or more cells of the accumulator. Each cell represents a possible solution for the unknown parameters. The cell of the accumulator having the most votes is assumed to be the estimate for the unknown parameters. An example for clustering is the commonly used Hough transformation [Hough, 1962].

Both, RANSAC and clustering do not give an accurate solution, but both methods are very robust to outliers. Therefore, they are often used as rough estimates for the searched parameters. They detect outlier and they prepare for maximum likelihood or mean least square estimation.

3.4.2   Propagation of Uncertainty

In statistics, the uncertainty of a variable $\boldsymbol{x}$ such that $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{v})$ is propagated from the uncertainty of $\boldsymbol{v}$ is formally written as

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}} = \boldsymbol{J}\boldsymbol{\Sigma}_{\boldsymbol{vv}}\boldsymbol{J}^\mathsf{T}, \tag{3.26}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{vv}}$ denotes the covariance matrix of $\boldsymbol{v}$ and $\boldsymbol{J}$ is the Jacobian (matrix of first-order partial derivatives) for the transformation function $\boldsymbol{f}$.

3.4.3 Representation of Uncertain Entities

In general, we can distinguish Euclidean and homogeneous representations for points, lines, planes and other geometric entities. These representations for points, lines, planes in 2D and 3D that are meaningful for this work are presented. More on representation of uncertain entities can be found in Heuel [2002], Förstner [2004], Meidow et al. [2009a], etc.

Representation of Points and Planes

In 2D, uncertain points are represented in Euclidean space by a 2-vector and its covariance matrix as

$$\chi : \{\boldsymbol{x}, \boldsymbol{\Sigma_{xx}}\}, \tag{3.27}$$

where

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{\Sigma_{xx}} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 \end{bmatrix}. \tag{3.28}$$

Analogously, in Euclidean 3D space, points are represented as 3-vectors

$$\mathcal{X} : \{\boldsymbol{X}, \boldsymbol{\Sigma_{XX}}\}, \tag{3.29}$$

where

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \boldsymbol{\Sigma_{XX}} = \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1 X_2} & \sigma_{X_1 X_3} \\ \sigma_{X_1 X_2} & \sigma_{X_2}^2 & \sigma_{X_2 X_3} \\ \sigma_{X_1 X_3} & \sigma_{X_2 X_3} & \sigma_{X_3}^2 \end{bmatrix}. \tag{3.30}$$

In projective space, the same points are represented using homogeneous coordinates as 3-vectors in 2D and 4-vectors in 3D, so that

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ \overline{w} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_E \\ \overline{x_h} \end{bmatrix} \tag{3.31}$$

and

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ \overline{T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_E \\ \overline{X_h} \end{bmatrix}, \tag{3.32}$$

where $\boldsymbol{x}_E$ and $\boldsymbol{X}_E$ are the Euclidean parts and x_h and X_h the homogeneous parts. The uncertainty of the homogeneous points is represented by the covariance matrices such that

$$\boldsymbol{\Sigma_{xx}} = \begin{bmatrix} \boldsymbol{\Sigma_{xx}} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \tag{3.33}$$

and

$$\boldsymbol{\Sigma_{XX}} = \begin{bmatrix} \boldsymbol{\Sigma_{XX}} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}. \tag{3.34}$$

Planes are dual of a 3D point in the projective geometry; therefore, they are also represented as 4-vectors

$$\mathbf{A} = \begin{bmatrix} A \\ B \\ C \\ \overline{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_h \\ \overline{A_E} \end{bmatrix}. \tag{3.35}$$

However, the Euclidean part A_h and the homogeneous part $\boldsymbol{A}_E$ are differently defined. 2D lines are dual of a 2D point.

Representation of Lines

Euclidean representation for lines in 2D: A 2D line has 2 degrees of freedom and can be represented using the Hessian normal form with angle γ and distance d:

$$x \cos \gamma + y \sin \gamma - d = 0. \tag{3.36}$$

Here, d denotes the shortest distance from the line to the origin of the coordinate system, and γ denotes direction angle of the normal vector to the line (Fig. 3.3b); therefore, an uncertain 2D line can be represented as:

$$\ell : \{\boldsymbol{h}, \boldsymbol{\Sigma}_{\boldsymbol{hh}}\}, \tag{3.37}$$

where

$$\boldsymbol{h} = \begin{bmatrix} \gamma \\ d \end{bmatrix}, \boldsymbol{\Sigma}_{\boldsymbol{hh}} = \begin{bmatrix} \sigma_\gamma^2 & \sigma_{\gamma d} \\ \sigma_{\gamma d} & \sigma_d^2 \end{bmatrix}. \tag{3.38}$$

A 2D line can be also represented using the slope-intercept form

$$y = mx + b. \tag{3.39}$$

Homogeneous representation for lines in 2D: In the projective geometry, a 2D line can be represented by

$$\ell : \{\mathbf{l}, \boldsymbol{\Sigma}_{\mathbf{ll}}\}, \tag{3.40}$$

where

$$\mathbf{l} = \begin{bmatrix} \mathbf{l}_h \\ \overline{l_0} \end{bmatrix} = \begin{bmatrix} a \\ b \\ \overline{c} \end{bmatrix}. \tag{3.41}$$

Here, $\mathbf{l}_h$ is the homogeneous part, $\mathbf{l}_0$ is the Euclidean part, and the condition $a^2 + b^2 + c^2 \neq 0$ has to be fulfilled[*]. In case $a^2 + b^2 = 0$, the obtained line is a line at infinity. $\mathbf{l}_h$ can be expressed using angle γ and $\mathbf{l}_0$ using distance d from (3.38) as

$$\begin{bmatrix} a \\ b \\ \overline{c} \end{bmatrix} = \lambda \begin{bmatrix} \cos \gamma \\ \sin \gamma \\ -d \end{bmatrix}, \tag{3.42}$$

with real factor $\lambda \neq 0$. Accordingly the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{ll}}$ can be derived as $\boldsymbol{\Sigma}_{\mathbf{ll}} = \mathbf{J}_h \boldsymbol{\Sigma}_{\boldsymbol{hh}} \mathbf{J}_h^T$ using

$$\boldsymbol{J}_h^\mathsf{T} = \frac{\partial \mathbf{l}}{\partial \boldsymbol{h}} = \begin{bmatrix} -\lambda \sin \gamma & 0 \\ \lambda \cos \gamma & 0 \\ 0 & -\lambda \end{bmatrix}. \tag{3.43}$$

The above derivation is useful for image lines extracted using methods such as Hough transformation; however, in computer vision, line segments defined by two points are often observed. A 2D line can be constructed as a join of two 2D points $\mathbf{x}_1$ and $\mathbf{x}_2$ using

$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2 = \mathsf{S}(\mathbf{x}_1)\mathbf{x}_2 = -\mathsf{S}(\mathbf{x}_2)\mathbf{x}_1, \tag{3.44}$$

where S is the skew-symmetric matrix

$$\mathsf{S}(\mathbf{x}) = \begin{bmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \tag{3.45}$$

---

[*]Homogeneous lines are dual to homogeneous points, but their homogeneous and Euclidean part are defined differently. Compare with the representation of 2D points in (3.31).

The covariance matrix for $\mathbf{l}$ is calculated as

$$\mathbf{\Sigma_{ll}} = \mathsf{S}(\mathbf{x}_2)\mathbf{\Sigma_{x_1x_1}}\mathsf{S}^\mathsf{T}(\mathbf{x}_2) + \mathsf{S}(\mathbf{x}_1)\mathbf{\Sigma_{x_2x_2}}\mathsf{S}^\mathsf{T}(\mathbf{x}_1), \tag{3.46}$$

assuming uncorrelated points $\mathbf{x}_1$ and $\mathbf{x}_2$.

Euclidean representation for lines in 3D: Typically, a 3D line is described by a direction vector $\boldsymbol{v}$ and a point $\boldsymbol{P}$. For this description, any point $\boldsymbol{P}$ belonging to the line can be used, thus there is more than one set of parameters describing one line. To solve this problem, Roberts [1988] introduced a line representation that is unique and unambiguous. This line representation was discussed, varied, and applied in photogrammetric context by Schenk [2004]. This line representation is based on two orientation parameters (α, θ) and two positional parameters (X_s, Y_s). The azimuth α and zenith θ can be deduced from the spherical coordinates of vector $\boldsymbol{v}$. (X_s, Y_s) are the coordinates of the intersection point with the plane X'Y', where X'Y'Z' is the rotated original coordinate system XYZ, so that the Z'-axis is parallel to the line (Fig. 3.3a). All equations required to calculate these parameters are given in Schenk [2004] and in Meierhold et al. [2008]. Each point on the line can be expressed as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_s \cos\alpha\cos\theta - Y_s\sin\alpha + t\cos\alpha\sin\theta \\ X_s\sin\alpha\cos\theta + Y_s\cos\alpha + t\sin\alpha\sin\theta \\ -X_s\sin\theta + t\cos\theta \end{pmatrix}. \tag{3.47}$$

Such a line is represented as

$$\mathcal{L}_{m_i} : \{\boldsymbol{L}_i, \mathbf{\Sigma_{L_iL_i}}\} \tag{3.48}$$

using a 4-vector $\boldsymbol{L}_i = [\alpha_i, \theta_i, X_{si}, Y_{si}]^\mathsf{T}$. This parameterization uses four parameters, which is the number of degrees of freedom of a 3D line.
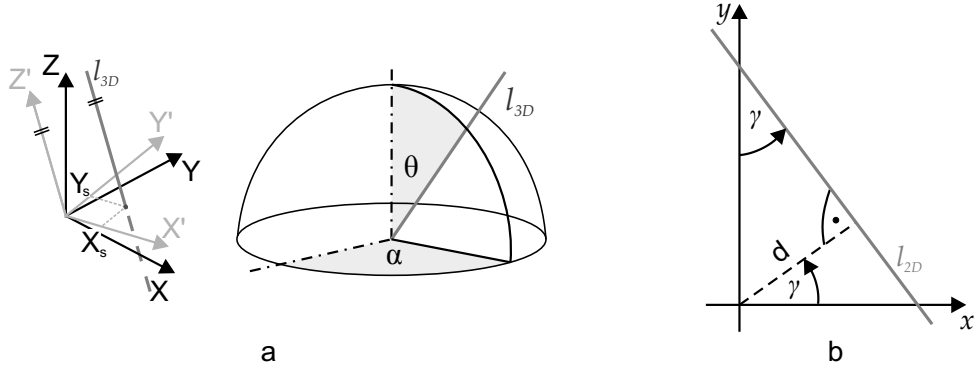

Figure 3.3: Parametrization of a line - graphical representation a) in 3D space using 4 parameters; b) in 2D space using 2 parameters.

Homogeneous representation for lines in 3D: Lines in 3D projective space are represented by homogeneous 6-vectors called also Plücker coordinates

$$\mathcal{L} : \mathbf{L} = \mathbf{X}_1 \wedge \mathbf{X}_2 = \begin{bmatrix} \mathbf{X}_2 - \mathbf{X}_1 \\ \mathbf{X}_1 \times \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_h \\ \mathbf{L}_E \end{bmatrix}, \tag{3.49}$$

where $\mathbf{X}_1$ is the join operator. In this representation $\mathbf{L}_h = \mathbf{X}_2 - \mathbf{X}_1$ is the homogeneous part, with two degrees of freedom, and $\mathbf{L}_E = \mathbf{X}_1 \times \mathbf{X}_2$ is the Euclidean part and can be interpreted

as normal of the plane $\mathcal{X}_1 \mathcal{X}_2 \mathcal{O}$ defined by $\mathcal{L}$ and the origin of the coordinate system $\mathcal{O}$. Plücker coordinates have to fulfill the condition

$$\mathbf{L}_h^\mathsf{T} \mathbf{L}_E = 0. \tag{3.50}$$

3.4.4 Euclidean and Spherical Normalization

The homogeneous representation is ambiguous due to the higher number of coordinates as degree of freedom. This ambiguity can be restricted by a normalization, for example by commonly used spherical normalization or Euclidean normalization. Euclidean normalization for homogeneous coordinates of a point is performed by

$$\mathbf{x}^\mathrm{e} = \mathrm{N}_\mathrm{e}(\mathbf{x}) = \frac{\mathbf{x}}{x_h}. \tag{3.51}$$

The covariance matrix of this point has to be also normalized by

$$\mathbf{\Sigma}_{\mathbf{xx}}^\mathrm{e} = \boldsymbol{J}_\mathrm{e}(\mathbf{x}) \mathbf{\Sigma}_{\mathbf{xx}} \boldsymbol{J}_\mathrm{e}^\mathsf{T}(\mathbf{x}), \tag{3.52}$$

where

$$\boldsymbol{J}_\mathrm{e}(\mathbf{x}) = \frac{1}{x_h^2} \begin{bmatrix} x_h \mathsf{I}_2 & -\boldsymbol{x}_E \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}. \tag{3.53}$$

Euclidean normalization for homogeneous coordinates of a line is calculated using

$$\mathbf{l}^\mathrm{e} = \mathrm{N}_\mathrm{e}(\mathbf{l}) = \frac{\mathbf{l}}{\|\boldsymbol{l}_h\|}. \tag{3.54}$$

The corresponding covariance matrix is then normalized by

$$\mathbf{\Sigma}_{\mathbf{ll}}^\mathrm{e} = \boldsymbol{J}_\mathrm{e}(\mathbf{l}) \mathbf{\Sigma}_{\mathbf{ll}} \boldsymbol{J}_\mathrm{e}^\mathsf{T}(\mathbf{l}), \tag{3.55}$$

where

$$\boldsymbol{J}_\mathrm{e}(\mathbf{l}) = \frac{1}{\|\boldsymbol{l}_h\|} \begin{bmatrix} \mathsf{I}_2 - \dfrac{\boldsymbol{l}_h \boldsymbol{l}_h^\mathsf{T}}{\|\boldsymbol{l}_h\|^2} & \mathbf{0} \\ -\dfrac{l_E \boldsymbol{l}_h^\mathsf{T}}{\|\boldsymbol{l}_h\|^2} & 1 \end{bmatrix}. \tag{3.56}$$

Spherical normalization is defined as

$$\mathbf{x}^\mathrm{s} = \mathrm{N}_\mathrm{s}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}. \tag{3.57}$$

In 2D, this normalization is equivalent for both: points and lines. Together with the entity, its covariance matrix has to also be normalized by

$$\mathbf{\Sigma}_{\mathbf{xx}}^\mathrm{s} = \boldsymbol{J}_\mathrm{s}(\mathbf{x}) \mathbf{\Sigma}_{\mathbf{xx}} \boldsymbol{J}_\mathrm{s}^\mathsf{T}(\mathbf{x}), \tag{3.58}$$

where

$$\boldsymbol{J}_\mathrm{e}(\mathbf{x}) = \frac{1}{\|\mathbf{x}\|} \left[\mathsf{I}_3 - \frac{\mathbf{x}\mathbf{x}^\mathsf{T}}{\mathbf{x}^\mathsf{T}\mathbf{x}} \right]. \tag{3.59}$$

3.4.5 Statistical Tests

As shown in Heuel [2002], we calculate the distance vector d and test the hypothesis H_0

$$H_0 : \boldsymbol{d} = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{y} = \boldsymbol{V}(\boldsymbol{y})\boldsymbol{x} = 0, \tag{3.60}$$

where x and y are the entities and U and V are the functions defining the relation between x and y. x and y are lines and we investigate the incidence in this particular case. Then we calculate the covariance matrix

$$\boldsymbol{\Sigma_{dd}} = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{\Sigma_{yy}}\boldsymbol{U}^\mathsf{T} + \boldsymbol{V}\boldsymbol{\Sigma_{xx}}\boldsymbol{V}^\mathsf{T}(\boldsymbol{y}). \tag{3.61}$$

Then H_0 is rejected with the significance level α if

$$T = \boldsymbol{d}^\mathsf{T}\boldsymbol{\Sigma_{dd}^{-1}}\boldsymbol{d} > \epsilon_H = \chi^2_{1-\alpha;n}. \tag{3.62}$$

4 Co-registration of 3D Building Models with TIR Imagery

The main goal of this thesis is to extract optimal thermal infrared (TIR) textures for a 3D building model, making in crucial to find the best fit between the existing 3D building model and the TIR images. In order to do this, an alignment between the model and image features is needed. This alignment is given by the projection matrix, including the exterior orientation parameters and the camera model. The exterior orientation parameters are typically taken from the GPS and INS navigation. The accuracy of the navigation system is, however, often not sufficient to provide an ideal match between the 3D model and the image. The alignment between the model and the image structures can be improved by matching the 3D model with the image features and re-calculating the exterior orientation parameters of the camera.

The exterior orientation parameters can be calculated by photogrammetric resection in space. In literature methods employing points, lines or both points and lines are used for this purpose. For buildings, linear structures are easier to detect in the images while the building corners (vertices), which can be represented by points, can be detected as intersections of lines. Therefore, in this work, linear structures in the image and in the model are utilized for the model-to-image assignment and to re-calculate the exterior orientation parameters and projection matrix.

In this chapter, a co-registration method based on a line-based model-to-image matching with tracking is presented. First, a methodology to match a single frame with a 3D building model is presented in Section 4.1. In this section, two alternative optimization methods for camera pose estimation based on the model-to-image correspondences are introduced. The first method is formulated in Euclidean space and uses RANSAC to revise the line-to-line correspondences and a Gauss-Markov model for the estimation. The second method is formulated in projective space and employs an accumulator approach to search for correspondences and a Gauss-Helmert model for the estimation. Both methods are implemented, tested, and compared. In Section 4.2, a strategy to support the matching using line tracking is sketched. Finally, in Section 4.3, the estimation method using the Gauss-Helmert model from Section 4.1 is extended with an outlier detection.

In this study, interior orientation of the thermal infrared (TIR) camera is assumed to be known. Also the uncertainties of the 3D building model and of the image features, as well as the accuracy of the navigation device, is assumed to be given. The uncertainties of the 3D building model and of the image features are considered in the estimation.

Parts of this chapter have been published in Iwaszczuk et al. [2010, 2012a,b,c, 2013a,b]; Iwaszczuk & Stilla [2014a,b,c].

4.1   Co-registration with a Single Image

For the co-registration of a single image and a 3D building model, two procedures need to be carried out: model-to-image matching and camera pose estimation. For model-to-image matching, corresponding features in the image and in the model have to be found. For this purpose, image features are extracted and selected. Extracted image features have particular reliability and geometric uncertainty, which is described in Section 4.1.1. For the matching, only the reliable features are taken. The geometric uncertainty influences the results of camera pose estimation and are taken into account for optimal pose estimation. Then, the 3D building models are projected into the image and the visibility is checked (Section 4.1.2). The 3D building models are treated as uncertain because of the inaccurate creation process and generalization; therefore, the uncertainty of the 3D building models is included in the pose estimation process as described in Section 4.1.3. Next, the line correspondences between the image and model lines are found and the optimal camera pose is calculated (Section 4.1.4).

4.1.1   Reliability and Uncertainty of Image Features

Edge extraction is carried out with the Canny edge detector [Canny, 1986]. This edge detection algorithm is based on the assumption that an edge is a border between two areas with different intensities in the image. The edge strength is calculated using a gradient. By varying the minimum edge strength required for a feature to be accepted as an edge during the extraction, different results are achieved. Lowering this parameter results in multiple detections because low-contrast edges are also included. However, this delivers more edges including noise. Setting the minimum edge strength to a high value results in the detection of "stronger" edges, but there may not be a sufficient amount of such edges for the matching. Thus, in this thesis, edge detection with varying minimum edge strength is proposed. As a result, we get three sets of detected edges. The edges are approximated with line segments for better handling in further steps. After this, very short line segments (≤ 3 [pix]) are removed and all detected sets of line segments are combined together.

It is assumed that building edges appear in the image as line segments. Short lines more likely correspond to other objects or to noise; therefore, all line segments are weighted using the minimum gradient strength and the length. The weights of the lines are calculated as follows:

$$g_j = \frac{1}{2}\left(\frac{l_j}{d_{max}} + \frac{a_j}{255}\right), \tag{4.1}$$

where g_j denotes the weight for the jth image line, l_j denotes the length of the jth line, $a_j \in [0,255]$ denotes the threshold for the minimum edge strength used for the extraction of the jth line, and d_{max} is the length of the diagonal of the entire image.

This weighting is used solely to identify the reliable edges and exclude those with low weights for the assignment of correspondences. For optimal pose estimation, the geometric uncertainty of extracted image features is needed. This value is derived from the covariances of the estimated parameters in the line segment approximation process. If the covariances are not available but the geometric uncertainty of the end points is given, the covariance matrix for the image line parameters can be calculated using error propagation.

4.1.2 Visibility Check for Lines

In texture mapping, checking for visibility of the building model and its parts is a very important step. It can be distinguished between visibility check for points, edges, and planes of the model. In this work, visibility check for lines at the matching step and visibility check for planes at the texture extraction stage is needed.

For texture extraction, Früh et al. [2004] proposed visibility check for lines based on the depth image. In this method, visible lines are determined by storing the depth image together with the normal vectors of surfaces. Line segments are extracted based on the direction of the normal vector, which means line segments are detected where the direction of the normal vector changes. This method is simple and delivers fast results, but projected lines are less accurate, the link to the topology stored in the model is more complicated and the edges between surfaces which build an angle close to 180° cannot be detected. Thus, in this study, a vector based visibility check for lines is proposed and implemented.

The principle of visibility check for lines is presented in Fig. 4.1. First, every point (vertex) of the 3D building model $\boldsymbol{X}$ is transformed to the 3D Euclidean coordinate system with the origin in the projection center $\boldsymbol{O}$ and the Z-axis pointing in the viewing direction (optical axis of the camera). This coordinate system will be called the *camera coordinate system*. This is done by rotation with rotation matrix R and translation to $\boldsymbol{O}$ using $\boldsymbol{X_{cam}} = R(\boldsymbol{X} - \boldsymbol{O})$ (see 3.4). Next, the faces of the 3D model, represented as polygons, are sorted from closest to the farthest, taking the centroid of each polygon as reference. This sorting procedure is used to simplify and speed up the algorithm. Polygons, which are arranged closer to the camera, are first checked to see if they cover a particular line segment. Afterwards, all points are projected into the image using $\mathbf{x} = \mathsf{K} \begin{bmatrix} \mathsf{I}_3 & 0_{3,1} \end{bmatrix} \mathbf{X_{cam}} = \mathsf{K} R \begin{bmatrix} \mathsf{I}_3 & -\boldsymbol{O} \end{bmatrix} \mathbf{X} = \mathsf{P}\mathbf{X}$, (see 3.5 and 3.7) $\mathbf{X} = \begin{bmatrix} \boldsymbol{X}^T & 1 \end{bmatrix}^T$ and $\mathbf{X_{cam}} = \begin{bmatrix} \boldsymbol{X}_{cam}^T & 1 \end{bmatrix}^T$ and normalized using Euclidean normalization $\mathbf{x^e} = \mathrm{N_e}(\mathbf{x})$. In the rest of this Section, the index $^\mathbf{e}$ is omitted and all homogeneous coordinates are assumed to be Euclideanly normalized.

Projected 3D points are also the end points of the projected edges of the 3D model, so projected line segments are recovered. Every line segment defines a line. For all possible line pairs, $\mathbf{m}_1 \wedge \mathbf{m}_2$, intersection points $\mathbf{x}_\times$ are calculated as

$$\mathbf{x}_\times = \mathbf{m}_1 \times \mathbf{m}_2. \tag{4.2}$$

Then it is checked whether or not $\mathbf{x}_\times$ lies on both line segments corresponding to $\mathbf{m}_1$ and $\mathbf{m}_2$. In this manner, intersection points of all line segments are found. These intersection points split line segments into parts. The visibility for each of these parts is tested so that the partial occlusion of lines can be handled. In order to do this, a point on the partial line segment is selected, say middle point $\mathbf{x_m}$, and a ray from this point through the projection center $\mathbf{O_{cam}}$ is created. This ray is described by a 3D line

$$\mathbf{L} = \mathbf{O_{cam}} \wedge \mathbf{X_m} \tag{4.3}$$

joining the 3D points $\mathbf{O_{cam}} = \begin{bmatrix} \boldsymbol{O}_{cam}^T & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ and $\mathbf{X_m}$, where $\mathbf{X_m} = \begin{bmatrix} \mathbf{x_m}^T & 1 \end{bmatrix}^T$ is point $\mathbf{x_m}$ in the camera coordinate system. The ray $\mathbf{L}$ intersects the line that it belongs to in point $\mathbf{Z}$ and all other planes defined by faces of the 3D building model in $\mathbf{Z'}_i$ (exceptions are planes parallel to $\mathbf{L}$), where i is indexing over the faces, starting from the closest face. If $|\boldsymbol{OZ'}_i| < |\boldsymbol{OZ}|$ the line segment or the part of the line segment is occluded.
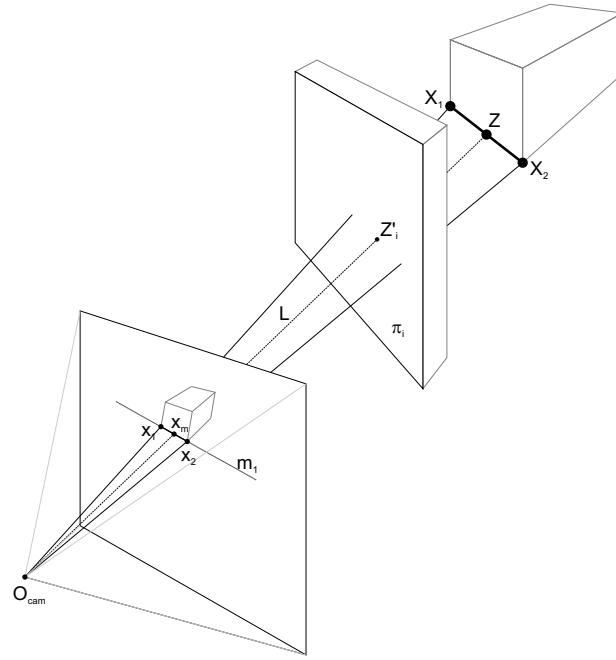
Figure 4.1: Visibility check principle

4.1.3   Uncertainty of 3D Building Models

The uncertainty of the 3D building models is related to the inaccuracies of creation and gener-alization. Many building models are created using aerial imagery where roof vertices and the height to the ground are measured. Often, the roof overlap is not modeled, and the wall edges are less accurate than the roof edges. Thus, in this thesis, different accuracies are assumed for the roof vertices and for the wall/ground vertices in the model. This is presented graphically as an error ellipses in Fig. 4.2. The Z-coordinate is assumed to be less accurate than the *X*-and *Y*-coordinates. Fig. 4.2 shows that, in the case of oblique airborne images, not only the *X*-and *Y*-components, but also the "Z"-component of the uncertainty, has a strong influence on the position error of the projected point.
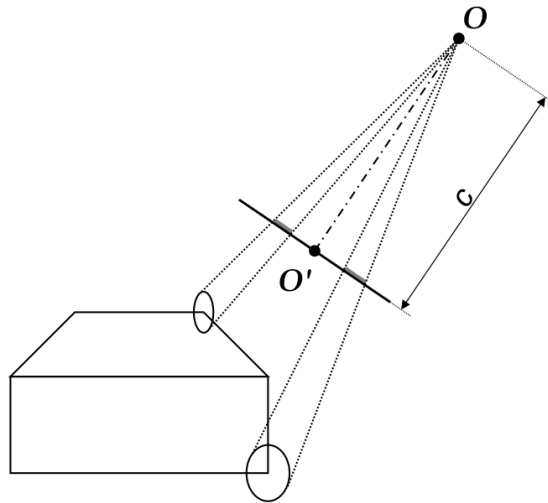
Figure 4.2: Projection of model point uncertainty into the image in oblique geometry.

The uncertainties of the parameters of the 3D line models (α_i, θ_i, X_{si}, Y_{si}) are not directly known in most cases. Usually, the position accuracy of 3D building models created from aerial imagery is given for building vertices; therefore, the uncertainty of model edges is treated as a consequence of the uncertainty of model vertices (Fig. 4.3) and is calculated using error propagation law as described in (3.26). The line parameters of the 3D model edges (α_i, θ_i, X_{si}, Y_{si}) are calculated from the model vertices $\boldsymbol{X}_{1i}$ and $\boldsymbol{X}_{2i}$. From this calculation, the Jacobian is retrieved and used for error propagation. The uncertainty of α_i, θ_i, X_{si} and Y_{si} is calculated and each model edge can be represented by a line $\mathcal{L}_{m_i}$ as described in (3.48). This representation is extended with the uncertainties of the line parameters as follows

$$\mathcal{L}_{m_i} : \{\alpha_i, \theta_i, X_{si}, Y_{si}\}, \boldsymbol{\Sigma_{L_i L_i}}. \tag{4.4}$$

The uncertainty of X_{Si} and Y_{Si} depends on the coordinate system; therefore, all calculations are carried out using the local coordinate system.
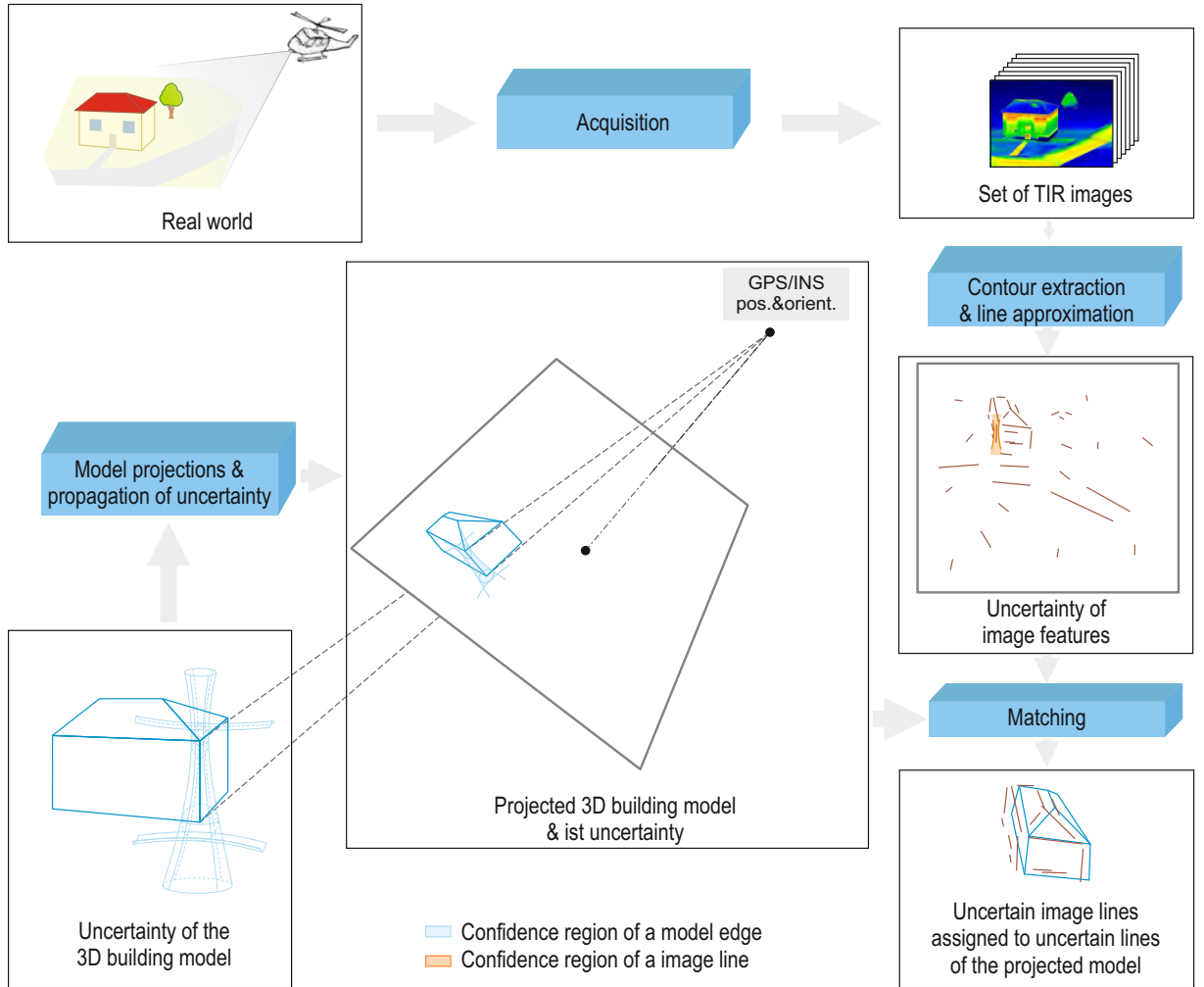

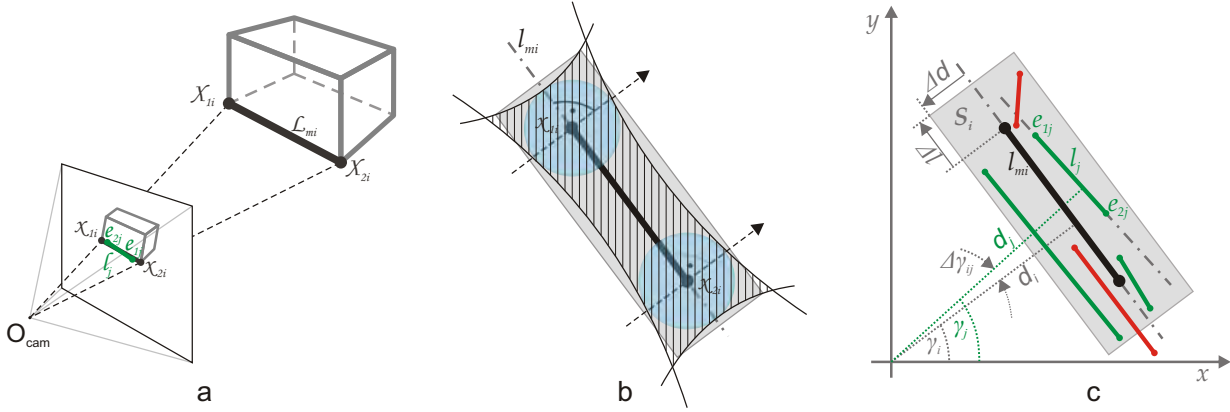Figure 4.3: Projection of model uncertainty into the image in oblique geometry.

Figure 4.4: Assignment of correspondences: a) Principle of line-to-line assignment due to the model projection; b) Graphical representation of the confidence region for end points (transparent blue), projected model edge (hatching), and its approximation with a rectangular buffer (grey); c) Assignment of line segments in 2D using buffer and angle approach. Symbol l_{mi} is the symbol i^{th} model line.

4.1.4   Model-to-Image Matching

In model-to-image matching, the correspondences between the model edges and the line segments detected in the image are searched. Using these correspondences, the camera pose is re-calculated. Once matched, the best fit between the projected 3D building model and the image is achieved.

Assignment of Correspondences using Buffer Approach

The assignment of corresponding lines is carried out in the 2D image space. The model lines are projected into the image using coarse exterior orientation parameters obtained by direct georeferencing. For each model edge, potentially corresponding image line segments are found. In this study, an assignment based on relative position and orientation is applied. First, a buffer notated as S_i, which exists around every projected visible model line segment (projected model edge), is calculated. The width of S_i is given by $2\Delta d$, where $\Delta d = 3\sigma_d$ and σ_d is the uncertainty of the parameter d (the distance from 3.36) of the projected model edge. σ_d is calculated by propagating the uncertainty of the camera position and the uncertainty of the model. For all image line segments within the buffer, the angle difference $\Delta\gamma_{ij}$ (Fig. 4.4) is calculated. All model line segments for which $\Delta\gamma_{ij}$ is smaller than a threshold $\Delta\gamma_{max}$ are accepted as correspondences, where $\Delta\gamma_{max} = 3\sigma_\gamma$ and σ_γ is the uncertainty of the parameter γ (angle from 3.36) of the projected model edge. The standard deviation σ_γ is calculated by propagating the uncertainty of the camera position and the uncertainty of the model. Propagation of uncertainty is carried out as described in Section 3.4.2. Formally, these conditions is written as follows:

$$l_j \leftrightarrow l_{mi} \text{ if } \{e_{1j}, e_{2j}\} \subseteq S_i \text{ and } |\Delta\gamma_{ij}| < \Delta\gamma_{max}, \tag{4.5}$$

where

$$\Delta\gamma_{ij} = \gamma_i - \gamma_j. \tag{4.6}$$

Here, l_{mi} denotes the ith model line, S_i is the search space for the ith model edge (buffer around l_{mi}), l_j denotes the jth image line segment, e_{1j} and e_{2j} are the end points of the jth image line segment, γ_i is the γ-parameter of the representation given by (3.36) i^{th} projected model edge, and γ_j is the γ-parameter (3.36) for the jth image line segment.
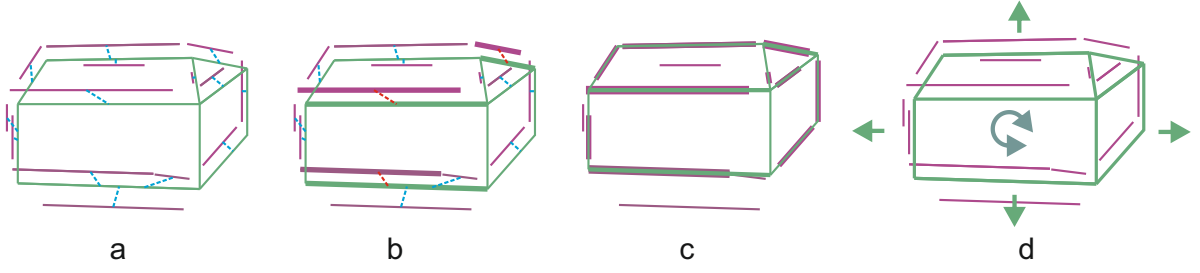
Figure 4.5: Outlier detectors: a) pre-selected line-to-line correspondences for RANSAC; b) randomly selected line-to-line correspondences for RANSAC; c) projected 3D building model after RANSAC and adjustment; d) principle of accumulator (arrows mean the movements of the projected 3D building model). Color coding: green - projected 3D building model, purple - image features, blue dashed line - preliminary line-to-line assignments, red dashed line - selected correspondences

Dealing with Outliers

These preliminary correspondences result in many outliers, even up to 80-90% [Iwaszczuk et al., 2012b]; therefore, these correspondences are reduced using an outlier detector. To achieve this goal, two methods, the RANSAC and the accumulator approach, are proposed, tested and compared.The main ideas of both methods in this application are presented in Fig.4.5.

Eliminating wrong assignments using RANSAC: RANSAC calculates the solution based on the minimal number of observations needed to solve the problem. This method is which is useful for data with high outlier rates. The theoretical background was described in Section 3.4.1.

Assigning correspondences using the buffer and the angle threshold allows for the selection of multiple image lines corresponding to one model line. This leads to many incorrect correspondences, which have to be eliminated or reduced. For this purpose, a RANSAC variation is implemented in this thesis. From the set of all hypothetical correspondences selected in the buffer and angle approach, we randomly select three correspondences from different parts of the model and calculate exterior orientation parameters without redundancy. We then check how many of the remaining correspondences fit with the randomly estimated exterior orientation parameters. This procedure is repeated k-times, and k is calculated as shown in (3.25). Here, 6 exterior orientation parameters are searched, thus $r = 6$, which means that three line correspondences are needed. The outlier rate ϵ is estimated as

$$\epsilon = N - N_{Mod}, \tag{4.7}$$

where N is the number of hypothetical correspondences selected by the assignment algorithm and N_{Mod} is the number of model lines which have at least one assigned image line. The RANSAC algorithm results in new exterior orientation parameters and in a set of correspondences, which are assumed to be correct. These data are taken as the input for the adjustment procedure described in Section 4.1.4.

Eliminating wrong assignments using accumulators: Clustering is a robust estimation method which gives reliable results even for applications with high outlier rates. The main idea of this commonly used method is described in Section 3.4.1.

Assuming the uncertainty of a few meters in the camera position and a few degrees in the camera orientation, the mismatch between the projected linear model structures and the structure in the image can be simplified to a 2D transformation. The error of the camera position and the

orientation results mostly in a 2D translation of the projected model, which can also result in a small rotation. To find the correct match between the projected 3D model and the image, a set of possible translations of the projected model is tested and the accumulator is filled, searching for the most likely position of the model in the image. The same process is repeated by rotating the projected 3D model using a small range of angles. This step results in a three dimensional accumulator filled with the number of correspondences assigned to every cell of the accumulator for every position and 2D rotation of the projected 3D model. Then, the maximum number of correspondences in the accumulator space are found and the assigned line-to-line correspondence are used to calculate the camera pose, as described in Section 4.1.4.

Optimal Pose Estimation

Using the selected correspondences, the camera pose is re-calculated to achieve a better fit between the projected model and the image features. Camera pose estimation is a calculation of six exterior orientation parameters $\widehat{\boldsymbol{\beta}} = \{\widehat{X}_0, \widehat{Y}_0, \widehat{Z}_0, \widehat{\omega}, \widehat{\phi}, \widehat{\kappa}\}$. These parameters can be estimated using line correspondences as observations. Fig. 4.4a shows the relation between the corresponding lines. The projected model edge has clearly defined endpoints that correspond to the model vertices: $\mathbf{X}_{1i} \leftrightarrow \mathbf{x}_{1i}$ and $\mathbf{X}_{2i} \leftrightarrow \mathbf{x}_{2i}$, where the symbol $\leftrightarrow$ indicates the correspondence relation.

The extracted edge in the image usually does not match the real edge because of the noise in the image and the edge detection techniques. Typically, an extracted edge does not have the same length as the corresponding real edge, so the building vertices do not correspond to the end points of the extracted edges $\boldsymbol{X}_{1i} \not\leftrightarrow \mathbf{e}_{1j}$ and $\boldsymbol{X}_{2i} \not\leftrightarrow \mathbf{e}_{2j}$. To handle this problem, the correspondence based on the linear features is used. Two relations can be formulated:

1. Co-planarity of $\mathcal{L}_{m_i}$ and ℓ_j,

2. Co-planarity of $\mathcal{X}_{1i}$, $\mathcal{X}_{2i}$ and ℓ_j.

Based on one of these relations, the optimal pose estimation can be solved in the Euclidean space or in the projective space.

Estimation in Euclidean space: For pose estimation in the Euclidean space, the first relation is used: co-planarity of $\mathcal{L}_{m_i}$ and ℓ_j. The perspective projection maps the 3D line parameters defined in (3.47) into the image. It maps a line in the 3D object space to a line (or in some exceptional cases to a point) in the image space. A line in the image space can be represented using (3.36).

For the perspective projection of points, the colinearity equations are used (3.1 and 3.2). These equations have been reformulated in Schenk [2004] and Meierhold et al. [2008] so that they can be applied for 2D line representation as shown in 3.39). In this thesis, further reformulation is conducted so that the collinearity equations apply for the 2D line representation from (3.36). Accordingly, the mapping function becomes

$$\boldsymbol{f} : \begin{cases} \gamma_j = f_1\left(X_0, Y_0, Z_0, \omega, \phi, \kappa, \alpha_i, \theta_i, X_{si}, Y_{si}\right) \\ d_j = f_2\left(X_0, Y_0, Z_0, \omega, \phi, \kappa, \alpha_i, \theta_i, X_{si}, Y_{si}, c_k, x_0, y_0\right) \end{cases} \tag{4.8}$$

where c_k is the camera constant and x_0, y_0 are the image coordinates of the principal point. Hence, the following functional model is defined by

$$\gamma_j + \widehat{v}_{\gamma j} = f_1\left(\widehat{X}_0, \widehat{Y}_0, \widehat{Z}_0, \widehat{\omega}, \widehat{\phi}, \widehat{\kappa}, \alpha_i, \theta_i, X_{si}, Y_{si}\right), \tag{4.9}$$

$$d_j + \widehat{v}_{dj} = f_2\left(\widehat{X}_0, \widehat{Y}_0, \widehat{Z}_0, \widehat{\omega}, \widehat{\phi}, \widehat{\kappa}, \alpha_i, \theta_i, X_{si}, Y_{si}, c_k, x_0, y_0\right), \tag{4.10}$$

where

$$\widehat{\boldsymbol{\beta}}_1 = [\widehat{X}_0, \widehat{Y}_0, \widehat{Z}_0, \widehat{\omega}, \widehat{\phi}, \widehat{\kappa}]^\mathsf{T} \tag{4.11}$$

is the vector of unknown parameters and

$$\mathbf{y}_1 = [\boldsymbol{\gamma}, \boldsymbol{d}]^\mathsf{T} \tag{4.12}$$

is the observation vector. After the linearization of the function $\boldsymbol{f}$, a Jacobian matrix

$$\boldsymbol{A_1} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\beta}_1} \tag{4.13}$$

is derived and the unknown parameters are calculated as

$$\boldsymbol{\beta}_1 = (\boldsymbol{A_1}^\mathsf{T} \boldsymbol{A_1})^{-1} \boldsymbol{A_1}^\mathsf{T} \mathbf{y}_1. \tag{4.14}$$

In order to introduce the uncertainty of both, the model and the image features, this functional model is extended with a stochastic model. The uncertainty of the 3D building models can be taken into account by extending the functional model with equations for the model line parameters as pseudo-observations $\mathbf{y}_2 = [\alpha_1, ..., \alpha_i, \theta_i, X_{si}, Y_{si}, ..., Y_{sn}]^\mathsf{T}$ such that

$$\alpha_i + \widehat{v}_{\alpha i} = f_3(\widehat{\alpha}_i) = \widehat{\alpha}_i, \tag{4.15}$$

$$\theta_i + \widehat{v}_{\theta i} = f_4(\widehat{\theta}_i) = \widehat{\theta}_i, \tag{4.16}$$

$$X_{si} + \widehat{v}_{X_{si}} = f_5(\widehat{X}_{si}) = \widehat{X}_{si}, \tag{4.17}$$

$$Y_{si} + \widehat{v}_{Y_{si}} = f_6(\widehat{Y}_{si}) = \widehat{Y}_{si}. \tag{4.18}$$

The vector of unknowns has also to be extended with the model line parameters. Accordingly, the Jacobian matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A_2} \end{bmatrix} \tag{4.19}$$

is extended with $4n$ observations and $4n$ unknowns, where n denotes the number of correspondences and

$$\boldsymbol{A_2} = \begin{bmatrix} \frac{\partial f_{3-1}}{\partial \alpha_1} & \cdots & & \cdots & \frac{\partial f_{3-1}}{\partial Y_{sn}} \\ \vdots & \frac{\partial f_{4-1}}{\partial \theta_1} & & & \vdots \\ & & \ddots & & \\ \vdots & & & \frac{\partial f_{5-n}}{\partial X_{Sn}} & \vdots \\ \frac{\partial f_{6-n}}{\partial \alpha_1} & \cdots & & \cdots & \frac{\partial f_{6-n}}{\partial Y_{sn}} \end{bmatrix} = \begin{bmatrix} 1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & 1 \end{bmatrix}. \tag{4.20}$$

This system of equations is solved using the Gauss-Markov model (Section 3.4.1) and the vector of unknowns

$$\widehat{\boldsymbol{\beta}} = [\widehat{X}_0, \widehat{Y}_0, \widehat{Z}_0, \widehat{\omega}, \widehat{\phi}, \widehat{\kappa}, \widehat{\alpha}_1, ..., \widehat{\alpha}_i, \widehat{\theta}_i, \widehat{X}_{si}, \widehat{Y}_{si}, ..., \widehat{Y}_{sn}]^\mathsf{T} \tag{4.21}$$

is estimated using

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{A}^\mathsf{T} \boldsymbol{P} \boldsymbol{A})^{-1} \boldsymbol{A}^\mathsf{T} \boldsymbol{P} \mathbf{y}, \tag{4.22}$$

where $\boldsymbol{P}$ is the weight matrix containing the predicted uncertainty of observations and pseudo-observations

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{y}_1 \mathbf{y}_1}^{-1} & \boldsymbol{0} \\ \boldsymbol{0}^\mathsf{T} & \boldsymbol{\Sigma}_{\mathbf{y}_2 \mathbf{y}_2}^{-1} \end{bmatrix}. \tag{4.23}$$

Here, $\boldsymbol{\Sigma}_{\mathbf{y}_1 \mathbf{y}_1}$ is the covariance matrix for the image line parameters $[\gamma_1, d_1, ..., \gamma_i, d_i, ..., \gamma_n, d_n]^\mathsf{T}$ and is usually derived from feature extraction as residual error of the line fitting. $\boldsymbol{\Sigma}_{\mathbf{y}_2 \mathbf{y}_2}$ is the covariance matrix of the line parameters $[\alpha_1, ..., \alpha_i, \theta_i, X_{si}, Y_{si}, ..., Y_{sn}]^\mathsf{T}$ of the 3D building model derived by error propagation as described in Section 4.1.3.

Estimation in projective space: Estimation of the exterior orientation parameters in the projective space is formulated using the co-planarity of l_j, $\mathcal{X}_{1i}$ and $\mathcal{X}_{2i}$. The co-planarity of l_j, $\mathcal{X}_{1i}$ and $\mathcal{X}_{2i}$ is expressed as incidence of the line l_j and the projected points represented as $\mathcal{X}'_{1i}$ and $\mathcal{X}'_{2i}$. The projected points $\mathcal{X}'_{1i} : \mathbf{x}'_{1i} = \mathsf{P}\mathbf{X}_{1i}$ and $\mathcal{X}'_{2i} : \mathbf{x}'_{2i} = \mathsf{P}\mathbf{X}_{2i}$, where P is the projection matrix. The incidence conditions $\mathbf{l}_j^T \mathbf{x}'_{1i} = 0$ and $\mathbf{l}_j^T \mathbf{x}'_{2i} = 0$ are written as

$$\mathbf{l}_j^{\mathsf{T}} \mathsf{P} \mathbf{X}_{1i} = 0, \tag{4.24}$$

$$\mathbf{l}_j^{\mathsf{T}} \mathsf{P} \mathbf{X}_{2i} = 0. \tag{4.25}$$

These two equations are directly adapted in the Gauss-Helmert model (Section 3.4.1) as conditions

$$\mathbf{g}_1(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{y}}) = \mathbf{l}_j^{\mathsf{T}} \mathsf{P} \mathbf{X}_{1i}, \tag{4.26}$$

$$\mathbf{g}_2(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{y}}) = \mathbf{l}_j^{\mathsf{T}} \mathsf{P} \mathbf{X}_{2i} \tag{4.27}$$

for the observations and parameters.

The uncertainty of the image features and the 3D building model can also be taken into account in the projective space. The covariance matrix for a 3D point $\mathcal{X}$ represented in homogeneous coordinates $\mathbf{X}$ can be directly derived from the cavariance matrix $\boldsymbol{\Sigma}_{XX}$ for the Euclidean representation $\boldsymbol{X}$ of this point as

$$\boldsymbol{\Sigma}_{\mathbf{XX}} = \begin{bmatrix} \boldsymbol{\Sigma}_{XX} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{4.28}$$

However, due to redundancy in the homogeneous representation, the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{XX}}$ is singular [Förstner, 2004], which leads to restrictions in the optimization. To solve this problem, all entities have to be spherically normalized [Kanatani, 1996], so that $\mathbf{l}_j^{\mathrm{s}} = \mathrm{N}_{\mathrm{s}}(\mathbf{l}_j)^*$, $\mathbf{X}_{1i}^{\mathrm{s}} = \mathrm{N}_{\mathrm{s}}(\mathbf{X}_{1i})$ and $\mathbf{Y}_{1i}^{\mathrm{s}} = \mathrm{N}_{\mathrm{s}}(\mathbf{Y}_{1i})$. In the rest of this section, the index $^{\mathrm{s}}$ is omitted, assuming the homogeneous coordinates are spherically normalized. This normalization has to also hold during the estimation as well, so the constraints

$$\mathbf{c}_1(\widehat{\mathbf{y}}) = \|\mathbf{l}_j\| - 1, \tag{4.29}$$

$$\mathbf{c}_2(\widehat{\mathbf{y}}) = \|\mathbf{X}_{1i}\| - 1, \tag{4.30}$$

$$\mathbf{c}_3(\widehat{\mathbf{y}}) = \|\mathbf{X}_{2i}\| - 1 \tag{4.31}$$

for the observations are needed.

To find the optimal solution for $\widehat{\boldsymbol{\beta}} = [\widehat{X}_0, \widehat{Y}_0, \widehat{Z}_0, \widehat{\omega}, \widehat{\phi}, \widehat{\kappa}]$, the optimization method for homogeneous entities presented in Meidow et al. [2009a] and Meidow et al. [2009b] is adapted for this functional model. The Lagrange function

$$\mathcal{L} = \frac{1}{2}\widehat{\boldsymbol{v}}^{\mathsf{T}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\widehat{\boldsymbol{v}} + \lambda_1^{\mathsf{T}}\mathbf{g}_1(\widehat{\boldsymbol{\beta}}, \mathbf{y}+\widehat{\boldsymbol{v}}) + \lambda_2^{\mathsf{T}}\mathbf{g}_2(\widehat{\boldsymbol{\beta}}, \mathbf{y}+\widehat{\boldsymbol{v}}) + \nu_1^{\mathsf{T}}\mathbf{c}_1(\mathbf{y}+\widehat{\boldsymbol{v}}) + \nu_2^{\mathsf{T}}\mathbf{c}_2(\mathbf{y}+\widehat{\boldsymbol{v}}) + \nu_3^{\mathsf{T}}\mathbf{c}_3(\mathbf{y}+\widehat{\boldsymbol{v}}) \tag{4.32}$$

is minimized, where λ and ν are the Lagrangian vectors. In contrast to Meidow et al. [2009a] and Meidow et al. [2009b], here the restriction for the estimated parameters $\mathbf{h}_1(\widehat{\boldsymbol{\beta}}) = 0$ is not needed, because the estimated parameters are defined directly as exterior orientation parameters X_0, Y_0, Z_0, ω, ϕ, κ. The observation vector for each pair of corresponding lines is written as $\mathbf{y}_{ij} = [\mathbf{l}_j, \mathbf{X}_{1i}, \mathbf{X}_{2i}]^{\mathsf{T}}$, where $\mathbf{l} = [a, b, c]^{\mathsf{T}}$ is the homogeneous representation for the image line segment and $\mathbf{X}_{1i}$, $\mathbf{X}_{2i}$ is the homogeneous representation of the vertices of the corresponding 3D building edge. The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{ll}}$ is assumed to be known as the result of the line fitting

---

*The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{ll}}$ calculated as shown in (3.46) is not singular but $\mathbf{l}$ is also spherically normalized in order to avoid ambiguity of the homogeneous representation.

or as the result of error propagation knowing the covariance matrices of the end points of the detected line segment (see 3.46).

The solution for the unknown parameters fulfilling (4.32) is calculated in a iterative manner, similar to what is described by Meidow et al. [2009b]. For all conditions, the Jacobians are calculated at the approximate values of $\boldsymbol{\beta}$ as

$$A = \begin{bmatrix} \frac{\partial \mathbf{g}_1(\boldsymbol{\beta},\mathbf{y})}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathbf{g}_2(\boldsymbol{\beta},\mathbf{y})}{\partial \boldsymbol{\beta}} \end{bmatrix}, \tag{4.33}$$

$$B = \begin{bmatrix} \frac{\partial \mathbf{g}_1(\boldsymbol{\beta},\mathbf{y})}{\partial \mathbf{y}} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{g}_2(\boldsymbol{\beta},\mathbf{y})}{\partial \mathbf{y}} \end{bmatrix}, \tag{4.34}$$

$$C = \begin{bmatrix} \frac{\partial \mathbf{c}_1(\mathbf{y})}{\partial \mathbf{y}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{c}_1(\mathbf{y})}{\partial \mathbf{y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{c}_1(\mathbf{y})}{\partial \mathbf{y}} \end{bmatrix}, \tag{4.35}$$

and used for normal equation

$$A^{\mathsf{T}} \left(B \Sigma_{ll} B^{\mathsf{T}} \right)^{-1} A \widehat{\Delta \boldsymbol{\beta}} = A^{\mathsf{T}} \left(B \Sigma_{ll} B^{\mathsf{T}} \right) a, \tag{4.36}$$

where

$$a = B C^{\mathsf{T}} \left(C C^{\mathsf{T}} \right)^{-1} \left(C \left(\mathbf{y} - \mathbf{y}^{(\tau)} \right) - c_{\tau} \right) - B \left(\mathbf{y} - \mathbf{y}^{(\tau)} \right) - g_{\tau}. \tag{4.37}$$

The index τ denotes the current iteration. Then the residuals are computed as

$$v^{(\tau)} = -\Sigma_{ll} B^{\mathsf{T}} \boldsymbol{\lambda} - C^{\mathsf{T}} \left(C C^{\mathsf{T}} \right)^{-1} \left(C \left(\mathbf{y} - \mathbf{y}^{(\tau)} \right) - \mathbf{c}_{\varnothing} \right). \tag{4.38}$$

with Lagrangians

$$\boldsymbol{\lambda} = \left(B \Sigma_{ll} B^{\mathsf{T}} \right)^{-1} \left(A \widehat{\Delta \boldsymbol{\beta}} - a \right). \tag{4.39}$$

Conditioning the coordinates: Switching from the Euclidean to the homogeneous representation for point χ in 2D or X in 3D is usually effected by adding 1 as an additional coordinate (homogeneous part). Hence, for a 2D point in Euclidean space $\boldsymbol{x} = [u, v]^{\mathsf{T}}$, the equivalent homogeneous representation is $\mathbf{x} = [u, v, 1]^{\mathsf{T}}$, and for a 3D point in Euclidean space $\boldsymbol{X} = [U, V, W]^{\mathsf{T}}$, the equivalent homogeneous representation is $\mathbf{X} = [U, V, W, 1]^{\mathsf{T}}$. In many photogrammetric applications, particularly in aerial photogrammetry, the points are given in geodetic coordinate systems (e.g., Gauss-Krüger, UTM), where the values for U and V is in the order of 10^6. Computations with such inconsistent numbers can cause the calculations to be numerically instable. To solve this problem, the homogeneous entities should be conditioned. Similar to the conditioning proposed by Heuel [2002], the entities are conditioned prior to the estimation by checking the condition

$$\max_{hE} = \max_i \left(\frac{\|x_{h_i}\|}{\|\boldsymbol{x}_{E_i}\|} \right) \geq f_{\min} = 0.1, \tag{4.40}$$

where x_{h_i} is the homogeneous and $\boldsymbol{x}_{E_i}$ the Euclidean part of a homogeneous entity $\mathbf{x}_i$ (see point representation from 3.31). If $\max_{hE} < f_{\min}$, the conditioning factor is calculated as

$$f = \max_{h0} \cdot f_{\min}. \tag{4.41}$$

If the Euclidean part $\boldsymbol{x}_E$ is much larger than the homogeneous part x_h, then f is calculated as shown in (4.41) can be smaller than the machine accuracy ϵ_h. Hence, if $f < \epsilon_h$ then f should be calculated as [Heuel, 2002]

$$f = \frac{f_{\min}}{\max_i(\|\boldsymbol{x}_{E_i}\|)}. \tag{4.42}$$

Next, each entity is conditioned using matrices

$$W_{\mathbf{x}}(f) = \begin{bmatrix} f I_2 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{4.43}$$

for the 2D points,

$$W_\mathsf{l}(f) = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & f \end{bmatrix} \tag{4.44}$$

for the 2D lines and

$$W_{\mathbf{X}}(f) = \begin{bmatrix} f I_3 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \tag{4.45}$$

so that the conditioned coordinates $\mathbf{x}^\mathsf{c}$, $\mathbf{l}^\mathsf{c}$ and $\mathbf{X}^\mathsf{c}$ are calculated as

$$\mathbf{x}^\mathsf{c} = W_{\mathbf{x}}(f_{\mathsf{im}})\mathbf{x}, \tag{4.46}$$

$$\mathbf{l}^\mathsf{c} = W_\mathsf{l}(f_{\mathsf{im}})\mathbf{l} \tag{4.47}$$

and

$$\mathbf{X}^\mathsf{c} = W_{\mathbf{X}}(f_{\mathsf{mod}})\mathbf{X}, \tag{4.48}$$

where f_{im} is the conditioning factor for the 2D image entities and f_{mod} is the conditioning factor for the 3D entities.

Conditioning entities causes a change in the transformation matrix. In this study, the transformation matrix is the projection matrix P, which can be reconditioned using

$$\mathsf{P} = W(f_{\mathsf{im}})^{-1}\mathsf{P}^\mathsf{c}W(f_{\mathsf{mod}}). \tag{4.49}$$

4.2   Coregistration with Image Sequences

Most VIS cameras, as well as the cameras operating in TIR domain, are able to capture image sequence with a relatively high frame rate. The frame rate of 20-25 frames per second is available in low and mid cost TIR cameras. Such frame frequency enables acquisition with a very large overlap between the images. Accordingly, the position shift in the image space from frame to frame is a few pixels for most objects. The viewing angle does not change between the frames significantly, so if the correct match is found in one frame, it is relatively easy to find correspondences in the next frame and calculate camera pose for this frame.

In an image sequence with a very large overlap between the frames, the whole process of model-to-image matching does not have to be carried out for all frames. In order to reduce computational effort, *key-frames* are used (Section 4.2.1) and selected lines are tracked from frame to frame (Section 4.2.2).

The general idea of the coregistration process is presented in Fig. 4.6. First, the current frame f_i, including its initial exterior orientation parameters, is selected, and edge detection in the image is carried out. These edges are used for line segment approximation, which results in a set of
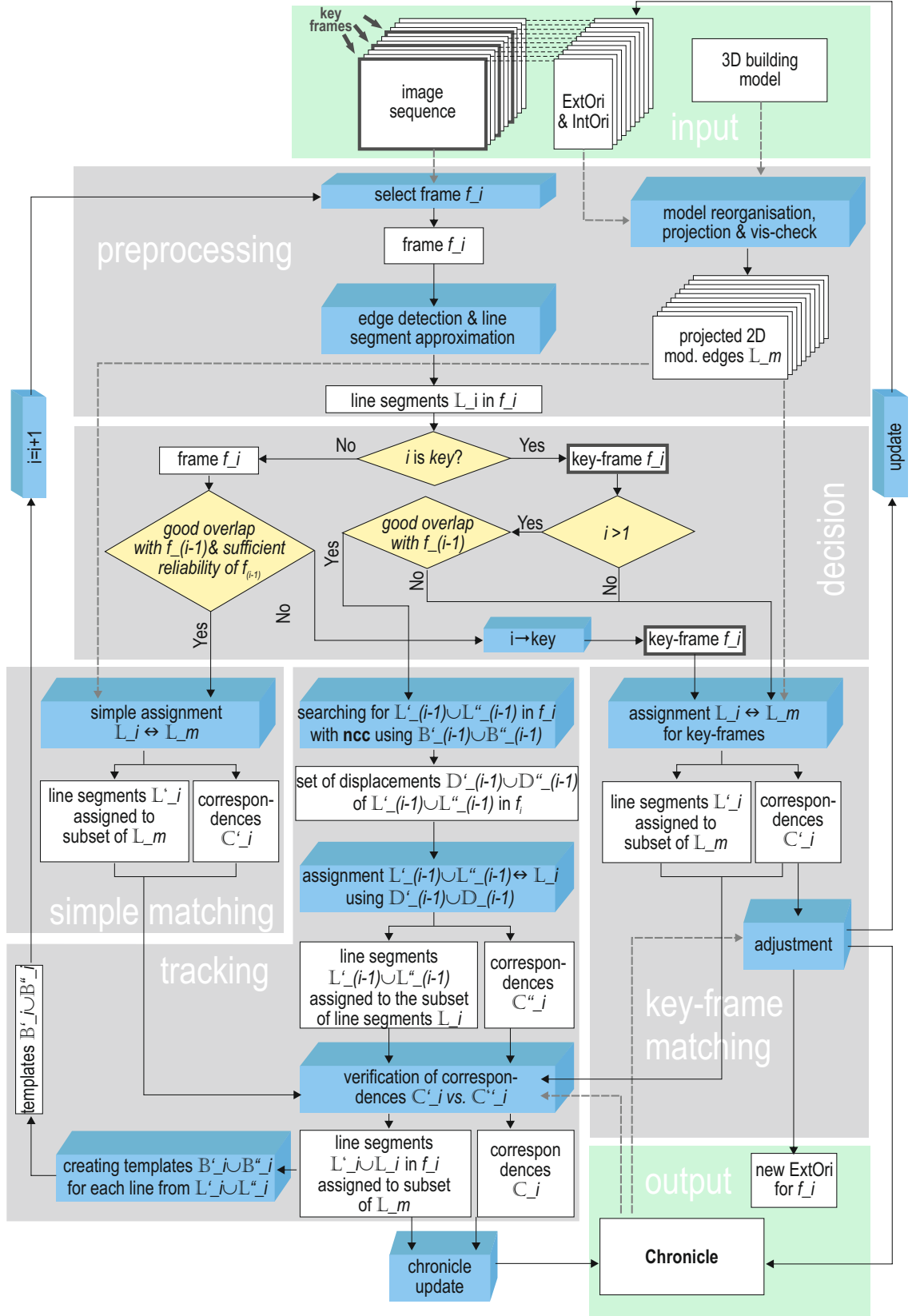
Figure 4.6: Algorithm for model-to-image matching using a video sequence and tracking of line segments.

image line segments $\mathbb{L}_i = \{l_{i,j}, \ j \in \{1, ..., j, ..., J\}\}$ in i^{th} frame. The edges of the 3D building model are contained in a set $\mathbb{L}_m = \{l_{m_n}, \ n \in \{1, ..., n, ..., N\}\}$. These edges are projected into the image and visibility is checked, which results in a set of 2D line segments in the image space.

Next, the decision level begins. Here, it is checked if the frame f_i is a *key-frame* or not. This decision stage is described in Section 4.2.1. Afterwards, depending on the result of the decision procedure, one or two of three following procedures are carried out:

- *Matching* for frame f_i, which is a *key-frames* (Section 4.2.1),

- *Simple matching* for frame f_i, which is not a *key-frame* (Section 4.2.1),

- *Tracking* for frame f_i, with good overlap with previous frame f_{i-1}(Section 4.2.2).

4.2.1   Key-Frame Solution

The main goals of the *key-frame* solution is to reduce computational effort and to ensure the reliability of the calculated camera pose for each frame. A *key-frame* is a frame in which the image-to-model matching and pose estimation are carried out as described in Section 4.1. In a *key-frame*, the chosen correspondences are selected independently of the previous frame. In general, the *key-frames* can be:

- *pre-defined* or

- *dynamically selected* during the process.

In order to initiate the process, the first frame f_i, $i = 1$ is always a *key-frame* (see Fig. 4.6). If the *key-frames* are pre-defined, they appear in certain intervals. The interval size should be adjusted to the overlap between the frames. For image sequences with a very high overlap the interval, can be higher than for frames with smaller overlap. If the overlap is not constant and not enough reliable correspondences with the model edges can be found, a dynamic selection of *key-frames* is applied.

Dynamic selection of *key-frames* is based on the current status of the reliability of matching and tracking. This reliability is the result of two main conditions:

- *sufficient overlap* between the frames f_i and f_{i-1},

- *sufficient reliability* of the assignment in f_{i-1}.

In a video sequence, the sufficient overlap between frames f_i and f_{i-1} is given in most cases. However, in some cases - if the camera is switched off for some time for example - the overlap can be too small to reliably track line segments from frame to frame. The reliability of the assignments depends on the number of selected correspondences and how much we believe that this assignment is correct. While the number of correspondences is simple to measure, the correctness of the assignment is more difficult to express.

As shown in Fig. 4.6 (*Key-frame matching* section), if the current frame f_i is a *key-frame*, for each line segment from $\mathbb{L}_m$, corresponding line segments from $\mathbb{L}_i$ are assigned. This assignment is done with the full matching procedure, including the elimination of wrong assignments and estimation. This procedure results in:

- a set of line segments $\mathbb{L}'_i = \left\{ l'_{i,k}, \ k \in \{1,...,k,...,K\} \right\}$ in the frame f_i, such that $\mathbb{L}'_i \subset \mathbb{L}_i$, which are assigned to the subset of projected model edges $\mathbb{L}_m$,

- a set of line correspondences $\mathbb{C}'_i = \{c_{i,k} = \{n,j\}, \ k \in \{1,...,k,...,K\}\}$

- new exterior orientation parameters for current frame f_i

As shown in Fig. 4.6 (*Simple matching* section), if the current frame f_i is not a *key-frame*, *simple matching* is carried out. In this procedure, the correspondences are selected based on statistical tests and the estimated displacement $\mathbb{D}_{i-1} = \{d_{i-1_n}, \ n \in \{1,...,n,...,N\}\}$ using

$$\text{mean}(\{d_{i-1_1},...,d_{i-1_n},...,d_{i-1_N}\}). \tag{4.50}$$

It results in:

- a set of line segments $\mathbb{L}'_i = \left\{ l'_{i,k}, \ k \in \{1,...,k,...,K\} \right\}$ in the frame f_i, such that $\mathbb{L}'_i \subset \mathbb{L}_i$, which are assigned to the subset of projected model edges $\mathbb{L}_m$,

- a set of line correspondences $\mathbb{C}'_i = \{c_{i,k} = \{n,j\}, \ k \in \{1,...,k,...,K\}\}$

In addition, if frame f_i has a sufficient overlap with previous frame f_{i-1}, tracking is carried out (Fig. 4.6, *Tracking* section). If frame f_i is a key frame, tracking is conducted for control; otherwise, it is used to find corresponding line segments between frames f_i and f_{i-1}, as well as calculate mean displacement (4.50).

4.2.2 Tracking Line Segments

Due to very small movements of the camera between the frames, line segments can be assumed to be only shifted by a few pixels in the next frame. They can be tracked, therefore, using cross-correlation (see 3.8). The cross-correlation method is suitable for tracking in this study because of the nearly invariant scale and because of viewing angle between two the neighboring frames. Accordingly, the appearance of the tracked line segment and its surrounding will stay almost unchanged.

The tracking with *key-frames* strategy gives five main possibilities regarding the verification of correspondences, matching, estimation, and connection between the frame. These five possibilities are presented schematically in Fig. 4.7. In the first option (Fig. 4.7a) matching and verification are carried out in *key-frames* only. After every matching, the camera pose is re-calculated for one frame. In other frames, only the line segments in the image are tracked in order to enable verification in the next *key-frame*. In the second option (Fig. 4.7b), the first case is extended with simple matching and verification in non-*key-frames*. Estimation is done for single *key-frames*, as seen in the first case in Fig. 4.7a. Conversely, the third option (Fig. 4.7c) calls for tracking, simple matching, and verification of correspondences, as well as the estimation in normal frames. The fourth option (Fig. 4.7d) makes it possible to connect the result of the matching in the *key-frames*. After every *key-frame*, a bundle adjustment with pose estimation is carried out, in which correspondences from previous *key-frames* are taken into account. This strategy can be also extended with estimation for every frame (Fig. 4.7e) so that correspondences from all previous frames are considered.

The weakness of the first option from Fig. 4.7a is that the tracking is based on cross correlation only and no verification of correspondences in non-*key-frames* is carried out. This strategy can be less reliable than the other options, since they allow the verification to be conducted in non-*key-frames*. The first strategy, however, is not computationally expensive. The second and third
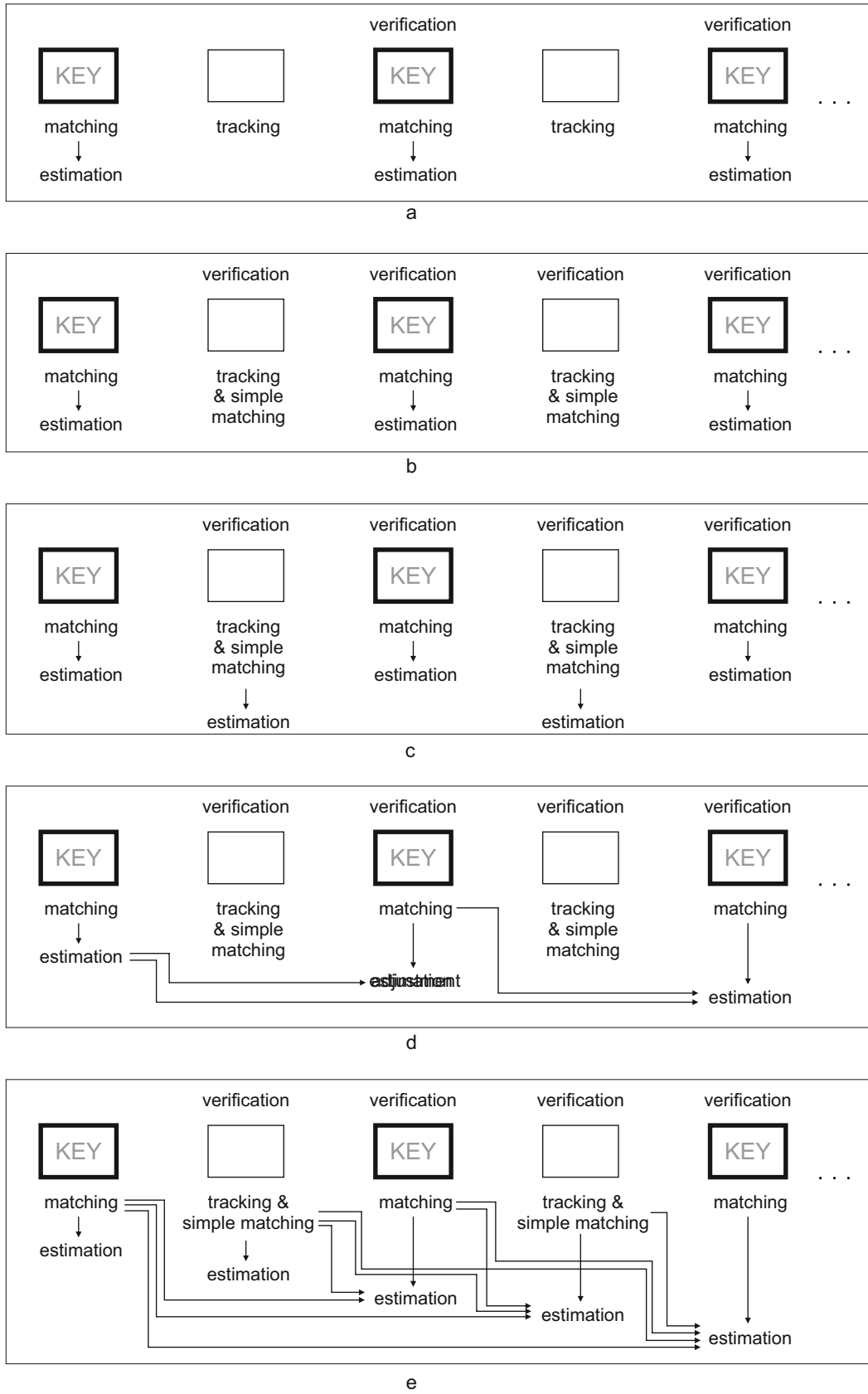
Figure 4.7: Possibilities for tracking and pose estimation using an image sequence

strategies (Fig. 4.7b and c) also use simple matching and verification after non-*key-frames*, but they do not apply bundle adjustment, which enables us to connect correspondences in all frames. The fourth and fifth strategies (Fig. 4.7d and e) use all correspondences in the bundle adjustment, but they can be computationally expensive, especially if conducted after every frame (Fig. 4.7e). In this study, the second and third strategies (Fig. 4.7b and c) are taken into further consideration because they are assumed to be reasonably balanced in terms of the reliability of the results and the computation time.

In Fig. 4.6 in the *Tracking* section, the tracking procedure is presented in detail. The process can be explained using three frames: f_{i-1}, f_i and f_{i+1}, where f_i is the current frame. In frame f_{i-1}, the subset $\mathbb{L}'_{i-1} \subset \mathbb{L}_{i-1}$ was assigned to a subset of model edges in simple matching procedure or in matching for key-frames. The set $\mathbb{L}''_{i-1} = \left\{ l''_{i-1,r}, \ r \in \{1, ..., r, ..., R\} \right\}$ is a set of image line segments traced from frame f_{i-2} and it was assigned to the subset of model edges in frame f_{i-1}. In frame f_i, correspondences $\mathbb{C}_i$ between detected image line segments $\mathbb{L}_i$ and tracked image line segments $\mathbb{L}'_{i-1} \cup \mathbb{L}''_{i-1}$ are found, using cross-correlation technique. For each line segment from $\mathbb{L}'_{i-1} \cup \mathbb{L}''_{i-1}$ in frame f_{i-1}, templates $b'_{i-1,k} \in \mathbb{B}'_{i-1}$ or $b''_{i-1,r} \in \mathbb{B}''_{i-1}$ are created and searched in frame f_i. This template is created as a few pixel buffer around the tracked line segment. The search is performed using normalized cross correlation. The correlation peak is searched by starting from the same position as in f_{i-1} and moving the template in all directions by few pixels. If the velocity model is known, the movements of the template is adjusted to this model. The position of the template at the correlation pick is assumed to also be the position of the tracked line segment $l'_{i-1,k} \in \mathbb{L}'_{i-1}$ or $l''_{i-1,r} \in \mathbb{L}''_{i-1}$ from f_{i-1} in f_i. From the correlation peak, the relative displacements $\mathbb{D}'_{i-1} \cup \mathbb{D}''_{i-1}$ are calculated.

Afterwards, image line segments from $\mathbb{L}_i$ corresponding to lines segments from $\mathbb{L}'_{i-1} \cup \mathbb{L}''_{i-1}$ are searched. This is done by projecting the line segments from $\mathbb{L}'_{i-1} \cup \mathbb{L}''_{i-1}$ into the frame f_i, adding the displacements $\mathbb{D}'_{i-1} \cup \mathbb{D}''_{i-1}$, and using statistical tests to find incident line segments from $\mathbb{L}_i$. This results in a set of tracked image line segments $\mathbb{L}'_{i-1} \cup \mathbb{L}''_{i-1}$ assigned to a subset of $\mathbb{L}_i$ via the correspondences $\mathbb{C}''_i$.

Then, the correspondences $\mathbb{C}''_i$ from tacking are verified with correspondences $\mathbb{C}'_i$ from the regular matching procedure or from the simple matching procedure, depending on the frame status (*key-frame*/normal frame). Tracked line segments from $\mathbb{L}'_{i-1} \cup \mathbb{L}''_{i-1}$ that are not assigned to any of the elements of $\mathbb{L}_i$ in frame f_i, become *virtual correspondences* for the model edges and are collected in $\mathbb{L}''_i$. This results in correspondences $\mathbb{C}_i = \mathbb{C}'_i \cup \mathbb{C}''_i$. For each line segment from $\mathbb{L}'_i \cup \mathbb{L}''_i$, templates $b'_{i,k} \in \mathbb{B}'_i$ or $b''_{i,r} \in \mathbb{B}''_i$ are created. These templates replace $b'_{i-1,k}$ and $b''_{i-1,r}$ from f_{i-1}, also for line segments from *virtual correspondences*, in order to avoid influence of geometric deformation. Templates $b'_{i,k}$ and $b''_{i,r}$ are used in frame f_{i+1} to continue the tracking.

All results of the tracking are stored in the chronicle. The chronicle entries are used to verify the correspondences and estimation. They document the process and can be retrieved throughout the whole procedure.

4.2.3 Visibility Prediction for Tracking

When tracking, some projected model edges cannot be visible at all times in the sequence. The information about model edge visibility in a particular frame is derived from the model and the approximated camera position. Whether or not the model edge is seen signals the state of a particular model edge in each frame. For each model edge, the following states are possible: alive - sound (fully visible), alive - injured (partially occluded), occluded (fully occluded but within the field of view), or dead (out of field of view). Each model edge can change its state if an event

occurs. Such events may occur for each model edge among the image sequence. Tab. 4.1 presents these events including the change of the state caused by each event.

Table 4.1: Possible events and states for tracked lines. Alive/sound suggest the edge is fully visible, alive/injured suggests it is partially occluded)

Event	Possible states before the event	Possible states after the event
Birth	$\varnothing$	Alive/sound Alive/injured
Appearing	Occluded	Alive/sound Alive/injured
Injury	Alive/sound	Alive/injured
Healing	Alive/injured	Alive/sound
Disappearing	Alive/sound Alive/injured	Occluded
Death	Alive/sound Alive/injured Occluded	Dead

The first event that occurs for a model edge is birth. It is the first moment, when the model edge is visible in the image. After birth, the model edge can have one of two states: alive/injured or alive/sound. Alive/injured means that the edge appears only partially in the frame or is partially occluded. This is the most common state directly after the birth of the edge. Rarely does an entire edge appear at once. If it were to occur it would directly result in an alive/sound state, which means the edge is fully visible edge. An alive/injured edge can become alive/sound during the healing event. Vice versa, an alive/sound edge can become alive/injured, if it gets partially occluded by an object or if part of the edge is not seen anymore in the current frame. Such an event is called injury. If the edge gets completely occluded by an object, such an event is called disappearing and it results in an occluded state. Disappearing can occur for alive/sound or alive/injured edges. The opposite of a disappearing event is an appearing event. It happens when an occluded edge becomes alive/sound or alive/injured. The last possible event is the death of the edge. It happens if the whole edge is no longer seen in the current frame, which means it is out of the field of view. Death can happen to an alive/sound, alive/injured, or occluded edge.

Defining the states of the model, makes it possible to know which edge's corresponding image line segments should be searched. Correspondences can be found only for alive edges. Injury is the only state which can be expressed with level of injury, it means how much of the edge is occluded. Highly injured edges are skipped when searching for correspondences.

4.3   Detection of Errors in the Observations

The inaccuracies in the model should result in large residuals in the estimation, which can be used to identify these inaccuracies. For this purpose, an outlier detector using a statistical test on corrections to the observations is implemented within the estimation. This is done by using a weight matrix [Niemeier, 2008]

$$\widehat{P} = \left(B\Sigma_{\text{ll}}B^{\mathsf{T}}\right)^{-1} \tag{4.51}$$

for computation of the weight coefficient matrix for the corrections to observations

$$Q_{\widehat{v}\widehat{v}} = \left(QB^{\mathsf{T}}\widehat{P}\left(\mathsf{I} - A(A^{\mathsf{T}}\widehat{P}A)^{-1}A^{\mathsf{T}}\widehat{P}\right)B\right)Q. \tag{4.52}$$

Then the vector of standardized corrections for the observations is calculated:

$$\psi_n = \frac{|v_n|}{\sigma_{v_n}} \tag{4.53}$$

where $v_n = y_n - y_{n0}$. Statistical tests are used to identify the outliers using the critical value

$$\epsilon_h = z_{1-\alpha}, \tag{4.54}$$

which makes it possible to reject a hypothesis with the significance level α. If $\psi_n > \epsilon_h$, then this observation is considered an outlier.

Observations identified as outliers are excluded from the estimation and the link to the corresponding model edges is stored. The procedure is repeated for every frame of the image sequence. Model edges that are frequently identified as outliers are labeled as possible errors in the 3D model.

Also, the reliability matrix $\boldsymbol{R}$ is calculated as

$$\boldsymbol{R} = \boldsymbol{Q}_{\widehat{\boldsymbol{v}}\widehat{\boldsymbol{v}}} \boldsymbol{Q}_{\mathrm{ll}}^{-1}. \tag{4.55}$$

The diagonal elements of the reliability matrix $\boldsymbol{r} = \mathrm{diag}(\boldsymbol{R})$ represent the redundancy components [Förstner, 1979] and the sum of these components is equal to the redundancy r of the equation system, meaning $\mathrm{tr}\,(\boldsymbol{R}) = r$. The redundancy components allow us to measure how strong the outlier is contained in the corresponding residual [Cothren, 2005].

5 Texturing with Thermal Data

In Chapter 4, the images were co-registered with the 3D building model. In this chapter, the process of texture mapping is continued, with the assumption that the camera pose is optimal with respect to the match between model and image features. In Section 5.1, the model representation in the context of texture mapping is discussed. Some typical geometric errors in the 3D building models are discussed and a strategy to deal with such problems is sketched. In Section 5.2, the approach applied for visibility check is briefly introduced, followed by a presentation of a workflow and a quality measure for the best texture selection from multiple images (Section 5.3). Afterwards, the method for local matching refinement is introduced (Section 5.4) and a strategy for texture creation combined with binary labeling for the invisible parts of the faces is shown (Section 5.5). Finally, the quality criteria used for evaluating of the texture mapping are defined (Section 5.6).

Parts of this chapter have been published in Iwaszczuk & Stilla [2010b,a] and [Iwaszczuk et al., 2015] .

5.1 Model Representation

The model representation is an important aspect for the texture extraction strategy. In some formats, e.g. *CityGML* [Gröger et al., 2012], the buildings are stored as polyhedral objects. This model representation makes it possible to store a building face as one geometric object and not as a collection of triangles. The texture can also be stored as one image for each face. The existing strategies for texturing triangulated 3D models, therefore, should be verified, and if necessary, modified.

3D building models can be represented using Constructive Solid Geometry (CSG) or Boundary Representation (B-rep). CSG is a method used for the 3D modeling of solids using solid primitives and boolean operations, which enable the modeling of complex surface. In B-rep, the boundary between the solid and non-solid is represented. As result, 3D models in this representation are collections of connected polygons. Which representation, CSG or B-rep, is more convenient depends on the application. For texturing, the B-rep is more practical because it is easier to assign textures to the polygons than to volumes.

5.1.1 Errors in Model Geometry

Presently, plenty of reconstruction software that enables semi automatic or even fully automatic generation of large datasets of 3D building models is available on the market [ESRI, 2015; SketchUp, 2015; virtualcitySYSTEMS, 2015]. The majority of building reconstruction software delivers B-rep models. During the reconstruction, however, CSG principles, including solid primitives and boolean operations, are used for modeling. As such, 3D building models are meant for visualization purposes only, thus the boolean operations are often not carried out correctly and the reconstructed building models include several volume intrusions (Fig. 5.1). Fig. 5.1a shows
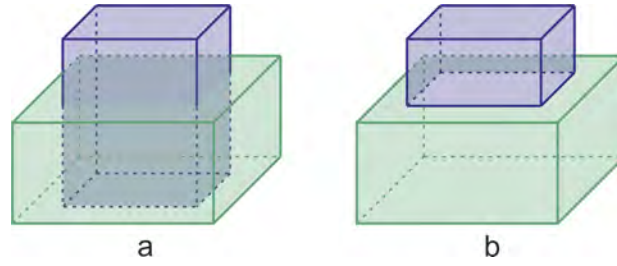
Figure 5.1: Volume intrusion: a) Volume intrusion caused by incorrect modeling; b) building modeled without intrusions.

a schematic example of volume intrusion. The complex structure of a building is modeled by placing a narrow and higher primitive (blue) in a wider and lower primitive (green). This way of modeling causes two main difficulties for texturing:

- The lower part of the higher primitive is invisible because it is inside another primitive. Also, part of the top polygon of the lower building is permanently invisible because it is inside of the higher primitive.

- Edges resulting from intrusion of two primitives are not explicitly included in the 3D model and cannot be used for matching.

Fig. 5.1b shows an example of correct modeling without intrusions. This is achieved by stacking two volumes.

Permanent invisibility: The permanent invisibility of some faces or their parts is challenging for automatic texture extraction. If invisible faces or their parts are not labeled, the system will try to find the texture for them. This can negatively affect the computation time. In addition, invisible faces downgrade the completeness of the texturing in the evaluation because they count as faces that could not be textured. To solve this issue, two approaches are possible:

- Invisible faces or their parts can be removed by 3D spatial queries. This strategy requires complex 3D operation and changes the original geometry.

- The invisible parts can be marked in the textures. For this purpose, binary textures can be created to permanently label the invisible parts (see Section 5.5). This can be done prior to the texturing process.

Missing edges: Missing edges, resulting from the primitive intrusion, are a problem for line base matching. Such edges are often detected in the image but cannot be correctly assigned to the model edges, where they are not explicitly modeled. This reduces the number of observations that can be used to calculate the optimal exterior orientation parameters. It also increases the risk of false assignments in the matching process. This problem can be solved by changing the geometry to the intrusion free B-rep (Fig. 5.1 b), or by computing the missing edges as intersections of 3D polygons. This topic, however, is not treated in this thesis and is mentioned only for the sake of completeness only.

Further problems with 3D geometry: A common error in 3D modeling is the opposite orientation of a normal vector for model surfaces. In computer graphics, the 3D geometry of a polygon is typically defined by points in clock-wise order; therefore, the normal vector of thi face is always directed to the observer. Storing 3D geometry in this way makes it possible to define the exterior and the interior of the building. Moreover, it allows us to use a back-face culling algorithm, which speeds up the visibility check. During the back-face culling, only the faces with a normal vector directed to the sensor are taken into consideration for the visibility check.

Fig. 5.2 shows an example of a 3D building model created manually with a commercial software and translated to the *CityGML* format. This model is affected by a few errors in 3D geometry. In addition to the already mentioned primitive intrusion (Fig. 5.2a), the normal vectors of its faces are randomly oriented (Fig. 5.2b), which hinders the identification of the interior, and exterior as well as the back-face culling. In Fig. 5.2c, the highlighted area is modeled as one face, while in reality, these are two independent walls that are not connected. This error is also the result of modeling for visualization purposes only. Finally, Fig. 5.2d shows how the primitives are not perfectly snapped. This error can result from inattentive modeling or numerical errors during the translation between the formats. Examples of another dataset with similar errors are presented in Fig. 5.3. Here, also, imperfect snapping causes small intrusions (Fig. 5.3a). Gaps between the buildings (Fig. 5.3b) can be seen.
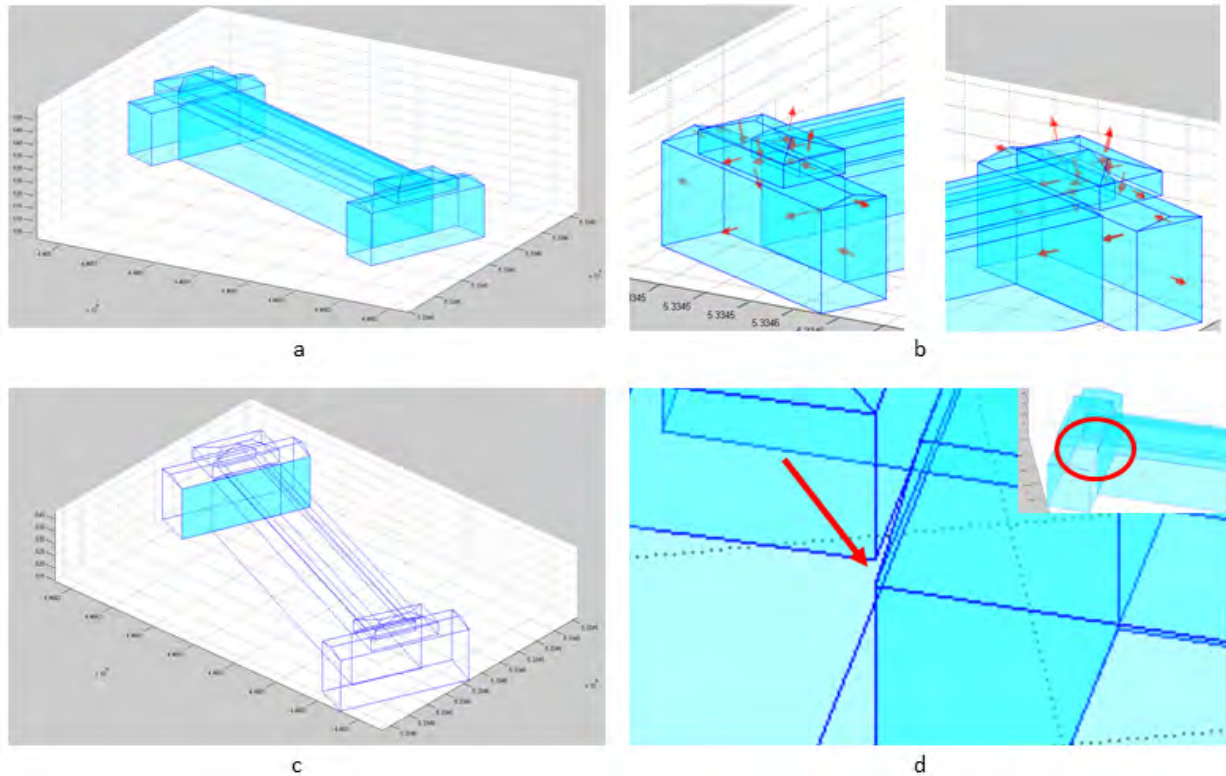
Figure 5.2: Examples of common modeling errors in 3D geometry of building models: a) intersecting primitives - the roof of the middle part on the building intersects with the roofs of the sides of the building; b) normal vectors are not correctly oriented; c) highlighted building surface is modeled as one face, while in reality these are two independent walls that are not connected; d) primitives are not snapped

Figure 5.3: Examples of errors in snapping: a) small intrusions; b) gaps between the buildings. This error can be the result of inattentive modeling or numerical errors during translation between the formats.

5.1.2   Representation of Geometry in *CityGML*

Considering the model representation and storage in *CityGML*, a question arises: What is the best way to model the geometry of complex building structures in a way best for texture mapping? This can be discussed using a simple example of two neighboring buildings. In general, there are nine possible ways to store the geometries of two neighboring buildings while considering the invisible part of both buildings. These nine possibilities are presented in Fig. 5.4. These building configurations are typical in city areas, as well as in residential districts, which are lined with houses and garages.
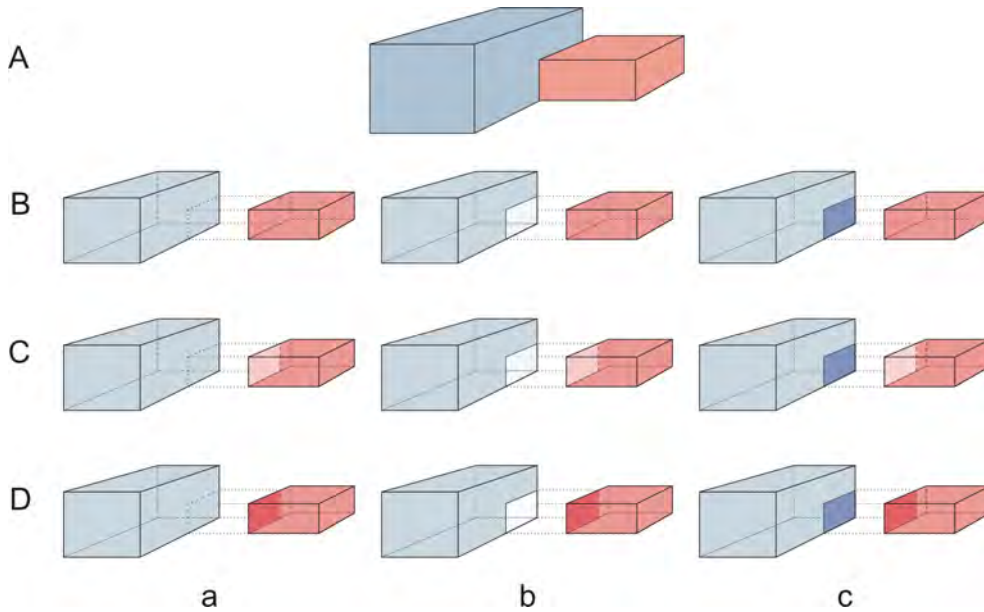
Figure 5.4: Possibilities for storing the geometry of two neighboring buildings with a common wall part: A) a building and an auxiliary building with a common wall part; B) the wall with the common part in the auxiliary building is stored as one polygon; C) the wall with the common part in the auxiliary building is stored as one polygon, where the common part is a hole; D) the wall with the common part in the auxiliary building is stored as two polygons, one for the common part and other for the rest of the building; a) the wall with the common part in the main building is stored as one polygon; b) the wall with the common part in the main building is stored as one polygon, where the common part is a hole; c) the wall with the common part in the main building is stored as two polygons, one for the common part and the other for the rest of the building.

Fig. 5.4A illustrates an auxiliary building, e.g. a garage (red), that was built touching the main building, e.g. a residential house (blue). According to Zlatanova et al. [2004], these buildings

fulfill a topological 3D relation *meet* where two faces, one from the garage and one from the house, partially occlude each other and have a common, invisible part. In a 3D building model, these faces can be stored in few different ways. In general, for each of the concerned faces, we can consider three cases:

- The face is stored as one polygon including the common part (in Fig. 5.4, column a for the residential house and row B for the garage).

- The face is stored as one polygon and the common part as a whole (in Fig. 5.4, column b for the residential house and row C for the garage).

- The face is stored as two polygons, the common part is a separate polygon (in Fig. 5.4, the column c for the residential house and the row D for the garage).

By combining these 3×3 possibilities, we get 9 different ways to store this model with two neighboring buildings (Fig. 5.4 B-a to D-c). In the order presented, it can be noticed that the cases on the main diagonal (B-a,C-b,D-c) treat the faces of both buildings in the same way. In case B-a, each face is represented by one polygon, including the common part. In this case, no topological relations can be modeled and the common part can not be explicitly identified, but the advantage of this representation is that removing one of these buildings does not require any changes in the geometry in the rest of the model. In case C-b, both faces are represented by one polygon with a removed common part. It suggests that the buildings are one entity and that there is a connection (passage) between them. In this case, after removing one of the buildings, a hole in another building occurs and the building has to be closed by changing the geometry of this face. However, such way of modeling is optimal for texture mapping because only the visible surface is included in the model geometry. In case D-c, each face is represented by two polygons - one for the invisible, common part and other for the visible part. Hence, the common part is stored using two polygons. Here, after removing one of the buildings, no change of geometry is required; however, the face of the remaining building is split into two polygons.

Cases C-c and D-b also illustrate an important method for storing the geometry. Here, the face of one building is represented by one polygon with a removed common part, while the face of another building is represented by two polygons. The advantage of these solutions is that the common geometry is stored only once. On the one hand, it reduces the size of the model because of the lower number of polygons; on the other hand, to which building the common polygon belongs to is not clear. To ensure that both buildings are closed, a link to the common geometry can be created as presented in Fig. 5.5. *CityGML* offers a possibility to use so called *XLinks*, to link a geometric element of one object. Fig. 5.6 shows how such common geometry can be stored only once. It shows two overlapping polygons stored in *CityGML* before and after their intersection. After the intersection, the common part is stored as part of one wall object and linked with the second wall object using a *XLinks*. The linked polygon has to get negative orientation to indicate that the lined geometry is oppositely directed. There are two possible outcomes. First, the common part can belong to the residential (main) building (Fig. 5.5 a). Second, this part can belong to the auxiliary building (Fig. 5.5 b). For texture mapping, it does not matter which solution is selected, but it can be important from the hierarchical point of view. The residential building can be seen as superior to the auxiliary building in the hierarchy. Accordingly, the common part should belong to the residential building. It can also be reflected in modeling using *Building* and *BuildingPart* [Gröger et al., 2012]. Löwner et al. [2012] recommend storing such surfaces as *ClosureSurface*.

In cases B-b and C-a in Fig. 5.4, the face of one building is represented by one polygon with a removed common part, and the face of another building is represented by one polygon that
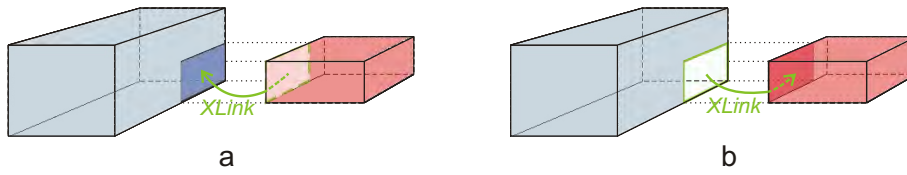
Figure 5.5: Storing geometry of neighboring building using XLinks: a) common part belongs to the main building; b) common part belongs to the auxiliary building.

includes the common part. This representation is inconsistent because it represents the building differently without any benefit and it is not favorable for any application. Similarly, cases B-c and D-a, where the face of one building is represented by two polygons, and the face of another building is represented by one polygon that includes the common part is also not advantageous in any task.

Fig. 5.6 presents the in which the common surface is stored in the *CityGML* code. Its geometry is stored in one building as a *SurfaceMember* and it is assigned to the second building using *XLink* and the ID from the first building. For texturing, this common geometry is excluded.
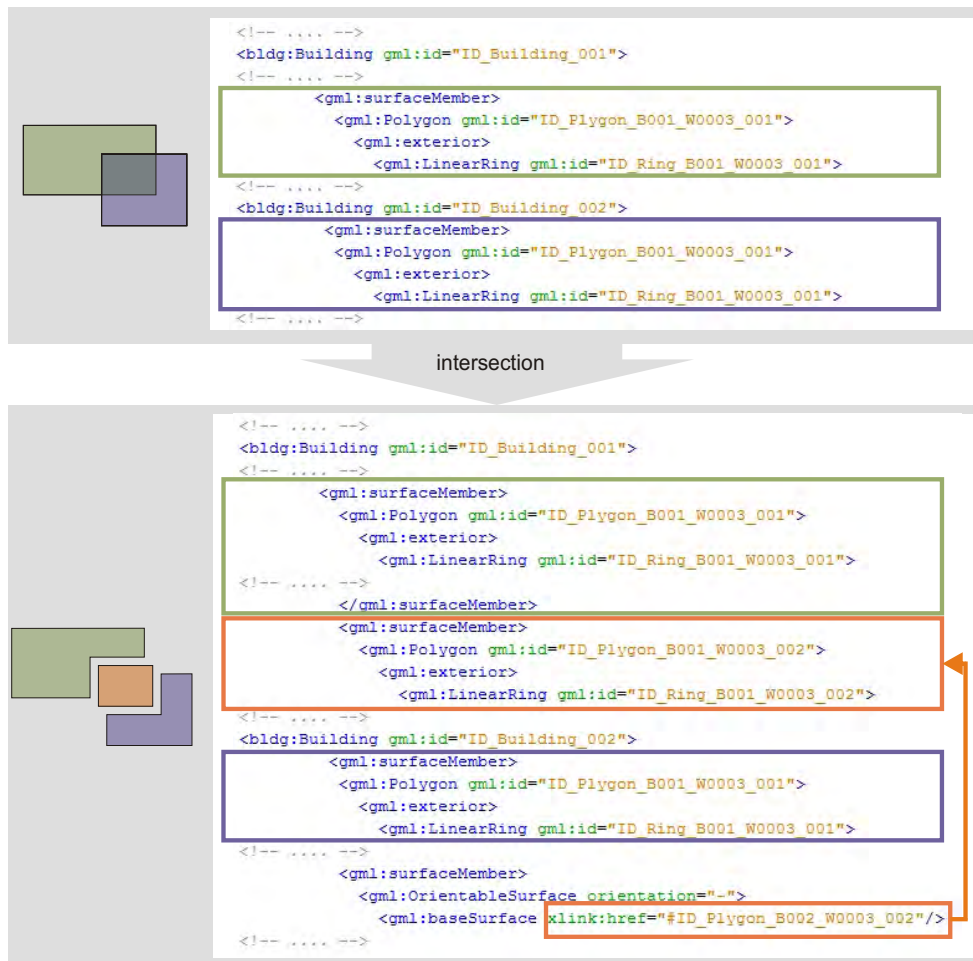
Figure 5.6: Geometry of neighboring building stored using XLinks in CityGML code

5.2   Visibility Check for Texture Extraction

Visibility check for texture extraction determines which pixels of the texture can be seen in a particular frame. This information is necessary to correctly extract parts of the image as textures and to determine the quality of a texture extracted from the particular frame. Unlike visibility check, when checking for lines (Section 4.1.2), a pixel-based approach is needed. For this purpose, the commonly used ray tracing technique is adapted. For each texture, each pixel is defined as a 3D point. This procedure also requires transforming the 3D model to the camera coordinate system and sorting the face,s as described in Section 4.1.2. Then in a similar way to (4.3), a ray **L** from the pixel to the projection center $\mathbf{O_{cam}}$ is created. Afterwards, on the way between $\mathbf{O_{cam}}$ and the object, **L** is intersected with all intermediate planes and investigated if the intersection point lies within the polygon. If so, the pixel is labeled as occluded. If no occlusion is detected, the pixel is labeled as visible. This algorithm is a variation of the ray tracing algorithm.

5.3   Best Texture Selection based on Acquisition Geometry

Best texture selection is needed when one face appears in multiple images. This is particularly important when working with image sequences taken at a high frame rate, higher than human perception can process (about 10-12 frames per second). In such cases, each face can be observed multiple times; therefore, for each of these of these occurrences, a quality measure has to be calculated and the best texture has to be selected.

For this purpose, a quality measure is used. The quality measure is defined as a function of occlusion; the angles between the normal of the investigated face and direction to the projection center, as well as its distance to the projection center.

The best texture selection procedure is schematically presented in Fig. 5.7. Starting from the first frame, each face p_j is projected into frame f_i and check if this face p_j was projected within f_i. If this condition is fulfilled, the visibility is checked (Section 5.2) and the occlusion rate o_{ij} is calculated. If p_j is visible in f_i, the quality q_{ij} for p_j in f_i is computed (Section 5.6.1). If $q_{ij} > q_{current,j}$, where $q_{current,j}$ is the quality of the current texture $t_{current,j}$ for f_i, texture $t_{current,j}$ is replaced by $t_{i,j}$. At the same time, the resolution map m_{ij} is created and the current resolution map $m_{currrent,j}$ is replaced by m_{ij}. This procedure is repeated for all I frames and J faces. The replacement of the texture and of the resolution map does not have to be calculated for each frame. Only the frame ID and image coordinates have to be stored. The final texture can be calculated after checking the whole sequence.

Effectively, this algorithm does not have to output the ready texture, only the reference for each face to the frame or frames where this face has the best quality. This procedure, therefore, can be done prior to the texture extraction or even prior to the data acquisition if the planned flying trajectory is known.

5.4   Refinement of Matching - Local Fitting

To get the best quality texture and reduce the influence of the errors in the building model, especially unmodeled roof overhang, a local fitting is implemented. Using the estimated exterior orientation of the camera (Section 4.1.4), the projected 3D building model is placed on the adjusted position in the image. To refine the fit, a local matching is applied for each edge independently. For this purpose, the gradient image is calculated and the 3D building model is projected into the gradient image. Around each edge, a buffer, including the pixels of the edge surrounding, is created. The size of the buffer has to be adjusted to the allowed movements,
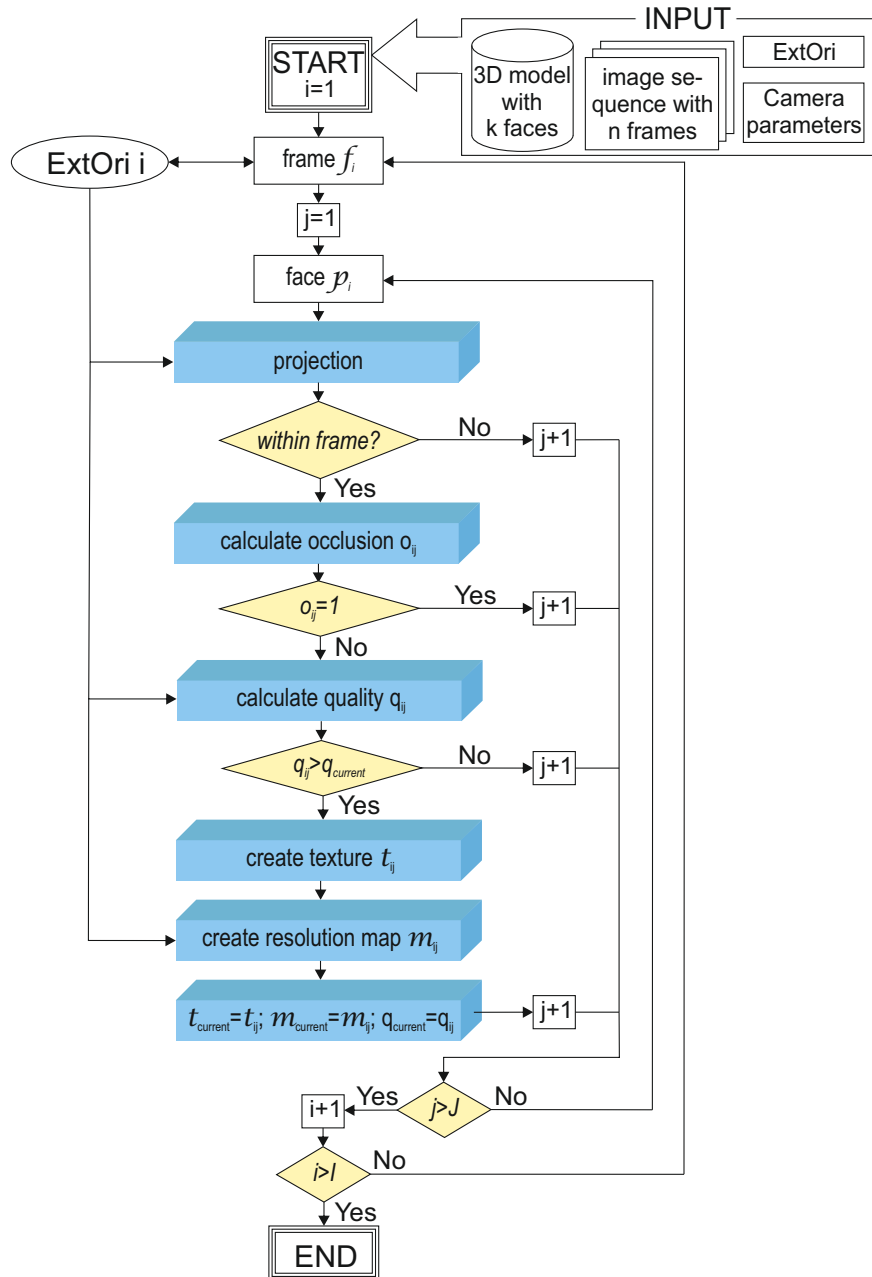
Figure 5.7: Workflow for best texture selection

which corresponds to the expected inaccuracy of the edge position in the image. The gradient values within the buffer are used to find the edge with a subpixel accuracy (Fig. 5.8). Instead of taking all pixels of the buffer, the points lying on the perpendicular lines at equal intervals are picked in order to reduce the computation time. Then, the least square method is applied to fit the line. The pixel coordinates of the picked points are set as observations and the gradient values are used as weights. Very small gradients are considered as noise, filtered using thresholding and set to the value 0 so that they do not influence the result of least square fitting.

Figure 5.8: Local refinement of the matching between the image and model edge using gradient values

5.5 Texture Extraction

In this thesis, a *texture* is defined as a rectangular image assigned to a polygon that depicts its appearance in a certain domain, e.g., visible (VIS) or thermal infrared (TIR). For non-rectangular textures, a bounding box around this face is created to achieve rectangular representation. These elementary textures can be stored as separate images or can be packed in a *texture atlas* [Früh et al., 2004; Kaul & Bohn, 2008], which is a patchwork of the elementary textures defined for each face. If the textures are extracted from image sequences and the best quality textures are selected, it is very likely that the elementary textures originate from different frames and , perhaps, have a different viewing angle. Often, if the textures are used to visually enrich 3D city models, the same texture is assigned to more than one face of similar buildings.

In this thesis, each elementary texture is stored separately; however, a texture atlas can be introduced in the future. Using the same texture for more faces based on building similarity is not possible because the textures are a source of information and are used for further processing and feature extraction. Besides the high geometric accuracy achieved by the local fitting (Section 5.4), attention is particularly payed to the occlusion. Two types of occlusion are defined:

- Permanent occlusion, resulting from the representation of the geometry (Section 5.1).

- Occlusion in each frame, caused by the objects placed between the camera and the inspected face (Section 5.2).

For each face, these two types of binary textures are generated. Permanent occlusion is determined before texturing, based on the geometry of the 3D model only. For this purpose, a pairwise checking is carried out if two faces are in the same plane. Then, one face is projected onto the second and vice versa, to label the occluded pixels.

Considering the errors in the geometry related to bad snapping (e.g. Fig. 5.3), a tolerance can be introduced while searching the touching faces. Faces lying in almost the same plane (angle between planes $\alpha < \alpha_{\max}$ or the distance between planes $d < d_{max}$) are also taken into account. For determining the intersection, a mean plane is created and the faces are projected on this plane

orthogonally. The occlusion in each frame is calculated based on the model geometry, the camera position, and the visibility check algorithm (Section 5.2).

The idea of the binary textures can be easily extended to label the various properties of the texture. Extrinsic objects, such as trees and cars, can cause occlusions, and as a result, pixels from the neighboring pixels for the visual purposes are interpolated. In this way, the textures can become a valuable source of information. Such multi-layer *Informative Textures* should be always created at the beginning of the texturing for each face.

Fig. 5.9 shows a workflow for creating a texture. First, a bounding box around the face is created and the permanent occlusion is labeled in the appropriate binary texture. This bounding box is then projected into the image and visibility is checked (a perfect match between the model and the image is assumed to simplify the presentation). After checking visibility, pixels occluded in the current frame are stored in another binary texture. Going from frame to frame, the binary texture for the occlusion in the binary texture with current occlusion is successively overwritten, while the binary texture with the permanent occlusion stays unmodified. This procedure results in the creation of three textures: two binary textures, including the occlusions, and one TIR texture. The occluded parts are blanked out in the TIR texture.
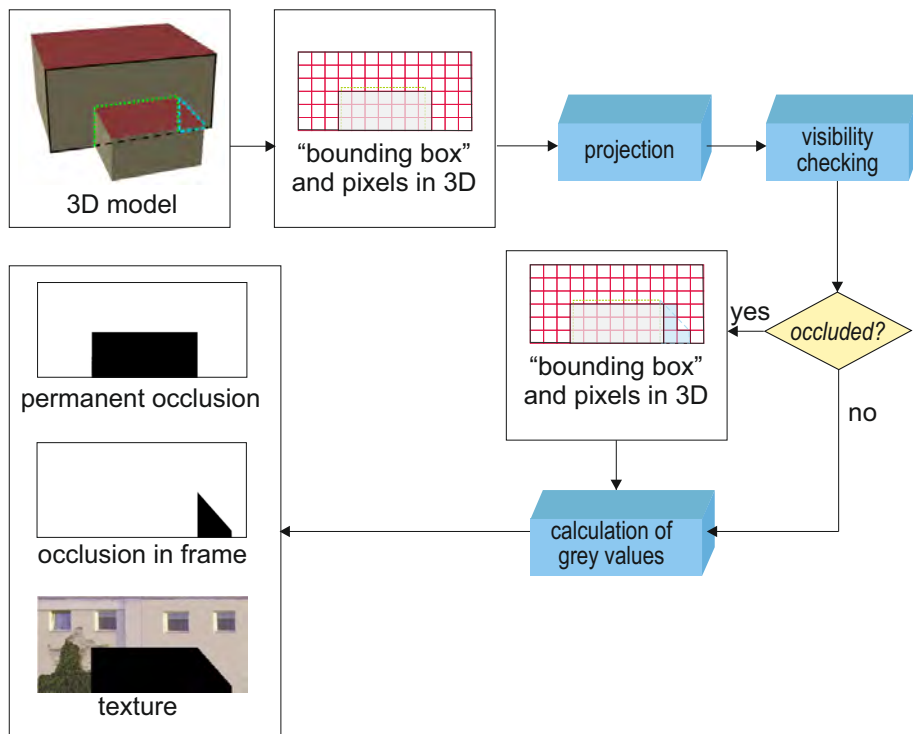
Figure 5.9: A workflow for texture extraction in one image

5.6   Quality Assessment of Extracted Textures

This section provides details on quality measures that are developed to assess the quality of texturing. Two groups of qualities are distinguished. First, geometric qualities, such as resolution and occlusion, are defined (Section 5.6.1). These qualities assess the geometric properties of the texture and give advice about the amount of information that can be extracted from the texture,

meaning it defines its level of details. Second, a matching quality is introduced (Section 5.6.2). This quality makes it possible to assess how well the extracted textures fit to the 3D model.

5.6.1 Geometric Quality Measures

The geometric qualities are qualities derived from the acquisition geometry, including the camera and object pose, as well as the inner orientation parameters of the camera. Therefore, these qualities can be used to assess the expected quality of textures in the planning stage or to assess the achieved textures after the flight.

Resolution

The texture's level of detail depends on its resolution. The resolution of 3D objects seen on the image plane is usually not unique along their surfaces. Unique resolution is possible only for planar objects that are parallel to the image plane. In nadir view photogrammetry, the ground resolution of the images is usually expressed using a ground sample distance (GDS), which is the distance between the pixel centers on the ground. It is calculated using the intercept theorem:

$$\frac{c_k}{s'} = \frac{H}{s},\tag{5.1}$$

where s is a distance on the ground, s' is its image in the sensor, c_k is the camera constant, and H is the flight height (see Fig. 5.10a). If $s' = 1pix$, then s is the ground sampling distance. Here, it is assumed that the ground is parallel to the sensor; therefore, all pixels have the same ground resolution. In oblique view, the GSD varies within the image significantly; it is smaller in the foreground and bigger in the background (Fig. 5.10b). The GSD does not give any information about the resolution of the 3D objects, such as façade or roofs, which is the most interesting aspect for texture mapping. Therefore, in this thesis, a local resolution for every object is defined as the length of a line segment placed on this object, which is depicted within one pixel. This line segment is parallel to one of the axes of the image coordinate system. Accordingly, two resolutions for one pixel can be calculated: in x- and in y-direction of the camera coordinate system.

An oblique view is equivalent to a nadir view of a sloped surface as shown in Fig. 5.10: Fig. 5.10b is equivalent to Fig. 5.10c.
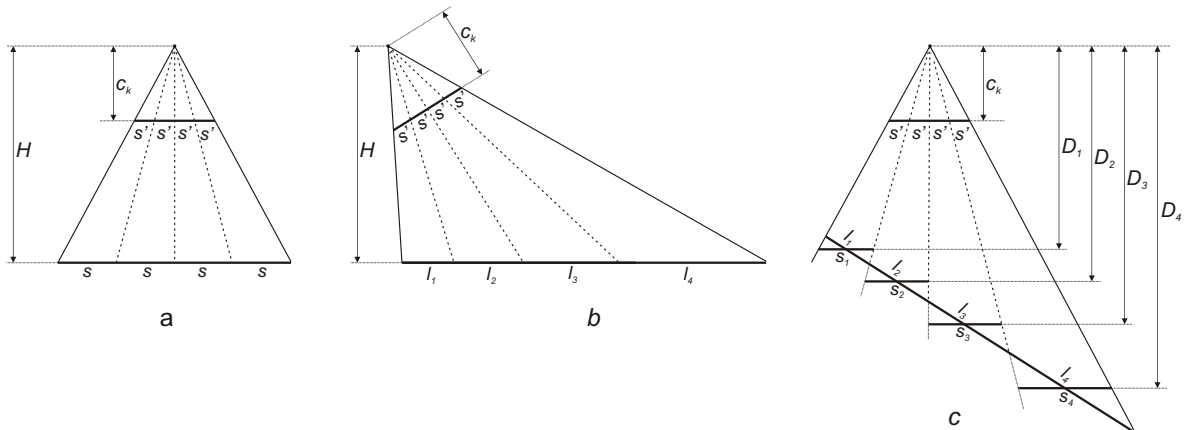
Figure 5.10: Geometry of nadir and oblique view: a) nadir view; b) oblique view ; c) nadir view of a sloped surface, which is equivalent to an oblique view of a flat surface.

This representation is suitable not only for the ground surfaces but also for other surfaces, e.g. façades or roofs. In this representation, a planar surface can be defined for each pixel. This

surface is parallel to the sensor and intersects with the photographed surface in the intersection point of the ray from the middle of the pixel with the photographed surface (Fig. 5.11a).
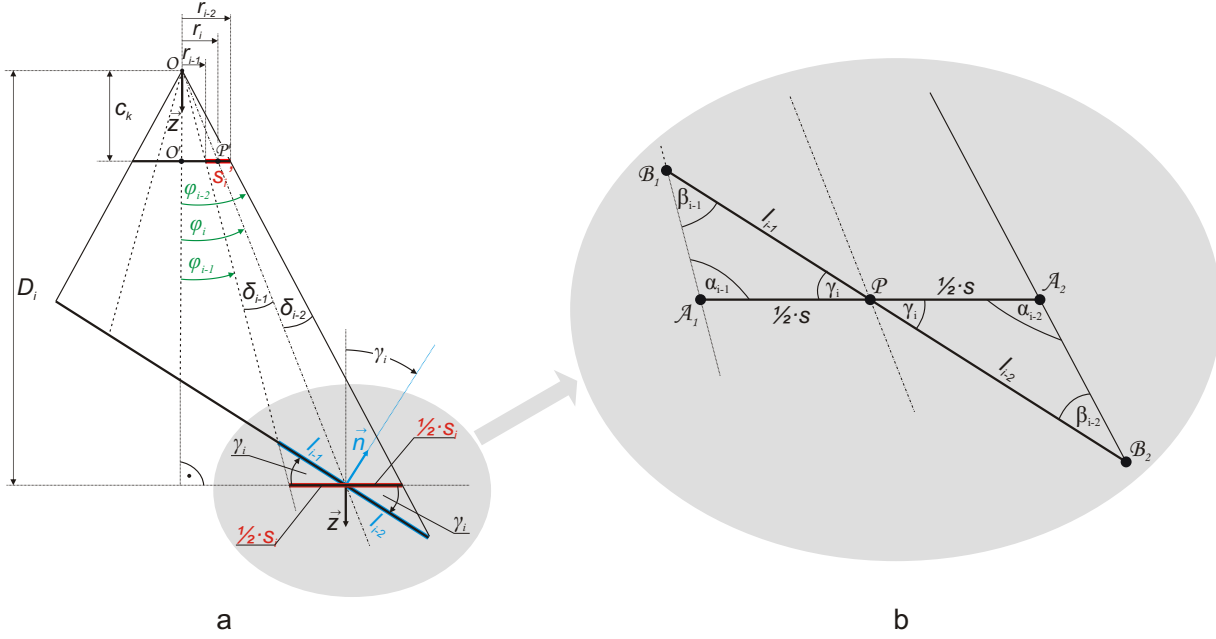

a   b

Figure 5.11: Detailed geometry for calculations of the resolution

If the distance D_i, which is the distance from the projection center to the photographed surface is known, the resolution of this parallel surface can be easily calculated using (5.1) by replacing H with D_i, which results in

$$\frac{c_k}{s'} = \frac{D_i}{s_i} \implies s_i = \frac{D_i s'}{c_k}. \tag{5.2}$$

Here, the index i denotes the pixel; however, in many cases, the photographed object is rotated by an angle

$$\gamma_i = \arccos(\frac{\overrightarrow{-z} \circ \overrightarrow{n}}{\|\overrightarrow{z}\| \|\overrightarrow{n}\|}), \tag{5.3}$$

where $\overrightarrow{n}$ is the normal vector of the photographed surface and $\overrightarrow{z} = [0, 0, 1]$. For every $\gamma_i > 0$, the length of the line segment on the photographed object is $l_i > s_i$. The ray from middle of the pixel does not intersect the line segment on the photographed object in the middle of this segment, but instead divides this segment into two line segments with the lengths l_{i-1} and l_{i-2} respectively (Fig. 5.11b). To calculate l_i, the triangles $\Delta A_1 B_1 P$ and $\Delta A_2 B_2 P$ should be solved. Using the Law of Sines, l_{i-1} is calculated from $\Delta A_1 B_1 P$

$$l_{i-1} = \frac{s_i \sin(\alpha_{i-1})}{2 \sin(\beta_{i-1})}, \tag{5.4}$$

where $\alpha_{i-1} = 180° - (90° - \phi_i) - \delta_{i-1} = 90° + \phi_i - \delta_{i-1}$ and $\beta_{i-1} = 180° - \gamma_i - \alpha_{i-1} = 90° - \phi_i + \delta_{i-1} - \gamma_i$. Similarly, l_{i-2} is calculated from $\Delta A_2 B_2 P$

$$l_{i-2} = \frac{s_i \sin(\alpha_{i-2})}{2 \sin(\beta_{i-2})}. \tag{5.5}$$

where $\alpha_{i-2} = 90° + \phi_i 0 \delta_{i-2}$ and $\beta_{i-2} = 90° - \phi_i - \delta_{i-2} - \gamma_i$. Here $\delta_{i-1} = \phi_i - \phi_{i-1}$ and $\delta_{i-2} = \phi_{i-2} - \phi_i$. The length l_i is calculated as the sum of l_{i-1} and l_{i-2}:

$$l_i = l_{i-1} + l_{i-2} = \frac{s_i}{2} \left(\frac{\sin \alpha_{i-1}}{\sin \beta_{i-1}} + \frac{\sin \alpha_{i-2}}{\sin \beta_{i-2}} \right). \tag{5.6}$$

ϕ_i is calculated by solving the triangle $\Delta OO'P'$ as follows

$$\tan \phi_i = \frac{r_i}{c_k} \implies \phi_i = \arctan \left(\frac{r_i}{c_k} \right). \tag{5.7}$$

Analogously,

$$\phi_{i-1} = \arctan \left(\frac{r_{i-1}}{c_k} \right) \tag{5.8}$$

and

$$\phi_{i-2} = \arctan \left(\frac{r_{i-2}}{c_k} \right). \tag{5.9}$$

If $s' = 1$ [pix], then δ_{i-1} and δ_{i-2} are very small angles. If we assume that $\delta_{i-1} \approx \delta_{i-2} \approx 0$, it implies that $\alpha_{i-1} \approx \alpha_{i-2} \approx 90° + \phi_i = \alpha_i$ and $\beta_{i-1} \approx \beta_{i-2} \approx 90° - \phi_i - \gamma = \beta_i$. Then l_i can be simplified to

$$l_i = s_i \frac{\sin \alpha_i}{\sin \beta_i} = \frac{D_i s' \sin \alpha_i}{c_k \sin \beta_i}. \tag{5.10}$$

Another simplification is presented in Fig. 5.12. Here l_i is length of the line segment, which has to be orthogonally projected onto the surface parallel to the sensor to fill one pixel

$$l_i = \frac{s_i}{\cos \gamma} = \frac{D_i s'}{c_k \cos \gamma}. \tag{5.11}$$
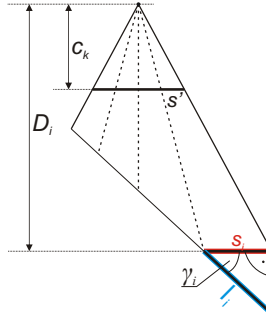
Figure 5.12: Simplified geometry for calculations of the resolution

Occlusion

Occlusion of a texture is a quality measure which is calculated based on acquisition geometry when considering self occlusion, or when considering extrinsic occlusion from additional data. This quality gives information about which percentage of the texture can be seen in a frame. Knowing the depth image of the scene, the occlusion factor o_{ij} is defined as

$$o_{ij} = \frac{n_{vis}}{N}, \tag{5.12}$$

where n_{vis} is the number of visible pixels in face, j in frame i and N is the number of pixels occupied by face j. The quality $o_{ij} \in [0, 1]$ takes value $o_{ij} = 1$ for fully visible textures.

General Geometric Quality

For best texture selection, which was described in Section 5.3, one significant quality measure is needed. Calculating the local pixel is computationally expensive; therefore, a simplified quality that takes resolution and occlusion into account is needed.

The more pixels in the texture are occluded, the lower the quality of the image. However, it is possible that a strongly occluded texture has a significantly higher resolution than the resolution of a completely visible texture. On one hand, if we want to extract a texture with the highest resolution, we should always select the parts of the texture with the highest resolution and combine them into one texture. On the other hand, we should keep in mind that every combination can cause small errors on the seam lines. Accordingly, an optimal balance between the occlusion and the resolution should be found using a quality measure

$$q_{ij} = \frac{a_1 o_{ij} + a_2 d_{ij} + a_3 \cos \gamma_{xij} \cos \gamma_{yij}}{a_1 + a_2 + a_3}, \tag{5.13}$$

where $a_1 + a_2 + a_3 \neq 0$. q_{ij} is computed for every face j in every frame i. γ_x, γ_y denote angles between the normal of a model polygon and the viewing angle of the camera, a_1, a_2, a_3 are coefficients, o_{ij} is the occlusion factor, and $d_i j$ denotes distance factor calculated by

$$d_{ij} = \frac{D_{max} - D_{ij}}{D_{max} - D_{min}}. \tag{5.14}$$

Here, D_{max} denotes maximum possible distance from the projection center to model points, D_{min} denotes minimum possible distance from the projection center to model points, and D_{ij} denotes the distance from the projection center to the center of a model polygon. For each face, a texture with the best quality q_{ij} is selected for texture mapping. In cases when a partially occluded face is selected for texturing, the missing part of it is searched in other frames, again considering their quality.

5.6.2   Matching Quality

Matching quality is introduced to measure how precisely the model was projected onto the image and how accurate the model is itself. Inaccuracies in data acquisition, the creation process, or the generalization can result in a faulty model. The matching quality of a texture t_{ij} assigned to face p_j in frame f_i is calculated using

$$\nu_{t_{ij}} = \sqrt{\frac{\sum_{k=1}^{K_j} \left(\frac{A_{ijk}}{l_{ijk_{IM}}} \right)^2}{K_j}}, \tag{5.15}$$

where A_{ijk} denotes the area between the projected model line segment and the actual corresponding line segment in the image, $l_{ijk_{IM}}$ denotes length of the projected model line segment, K_j denotes number of sides in the face polygon p_j (Fig. 5.13).

The actual corresponding line can be measured in the image by a human operator, or it can be estimated using the intensity values surrounding the projected model line segment in the image. To evaluate the texture extraction in a frame f_i, a quality measure

$$\nu_{f_i} = \frac{\sum_{j=1}^{J} \left(\nu_{t_{ij}} \right)}{J} \tag{5.16}$$
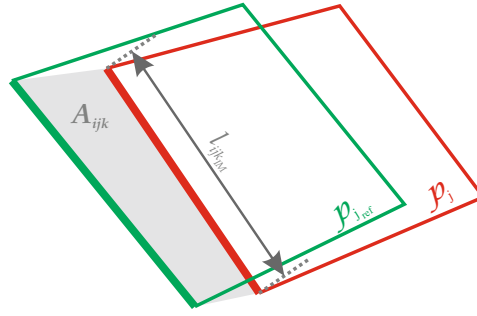
Figure 5.13: Calculation of matching quality

is used. To assess the matching quality for a face p_j among all frames

$$\nu_{p_j} = \frac{\sum\limits_{i=1}^{I} \left(\nu_{t_{ij}} \right)}{I} \tag{5.17}$$

is calculated. Combining (5.16) and (5.17), the matching quality ν of the whole matching process is computed as

$$\nu = \frac{\sum\limits_{i=1}^{I} \sum\limits_{j=1}^{J} \left(\nu_{Tij} \right)}{I \cdot J}. \tag{5.18}$$

In (5.16) - (5.18) J denotes the number of visible faces in a frame f_i, and I denotes the number of frames.

6 Window Detection in Thermal Façade Textures

In the previous chapter, it was explained how thermal textures are extracted from airborne TIR image sequences. In this chapter, the methodology for window detection in such textures is presented. First, the general problem related to window detection in TIR images is addressed (Section 6.1). Then, the design for a window model and façade model used for detection are described (Section 6.2). Within the method presented, a search for regions of interest is implemented (Section 6.3). These regions are used as candidates for windows to reconstruct the lattice of windows (Section 6.4).

Parts of this chapter have been published in Iwaszczuk et al. [2011a,b].

6.1 Problem Overview

In optical images recorded in the visible (VIS) domain, the appearance of objects depends on material properties, illumination, and the viewing angle. In thermal infrared (TIR) images, the appearance depends on the material and the viewing angle as well. However, with the exception of materials with specular (mirror-like) reflectivity, the viewing angle in TIR influences the appearance less than in the visible domain. Illumination is the main factor influencing appearance in the VIS domain, while in TIR images the radiation of objects depends on temperature distribution. Accordingly, the appearance of an object in TIR images depends on the temperature and the material of the object when imaging diffused surfaces (e.g., wood, plastic). It also depends on the viewing angle and the temperature of the surroundings when imaging specular surfaces (e.g., glass, aluminum).

Fig. 6.1 schematically shows what will happen when photographing a building façade from the air (Fig. 6.1a) and from the ground (Fig. 6.1a). The window panes reflect the thermal radiation of objects in the surroundings or of the sky. In the example presented in Fig. 6.1a, the top window reflects a tree with temperature t_1, the middle window reflects a car with temperature t_2 , and the lowest window reflects the ground with temperature t_3. In the example in Fig. 6.1b, the lowest window reflects a tree with temperature t_1, the middle window reflects a façade of an another building with temperature t_4 , and the top window reflects the sky with temperature t_5. Knowing the temperatures t_1, t_2,t_3, t_4 and t_5, corrections to the measured thermal radiation can be introduced. Knowing the reflectivity of the photographed object and the temperature of the reflected surface, it is possible to correct the measured thermal radiation to obtain the temperature of the measured surface. Knowing the 3D geometry of the photographed scene and the location of the windows in the façade makes it possible to correct the temperature of the windows for thermal inspection or to simply exclude them. Knowledge about the location of windows in the façade can also be helpful for identifying heat leakages, as heating systems are often placed below the windows.
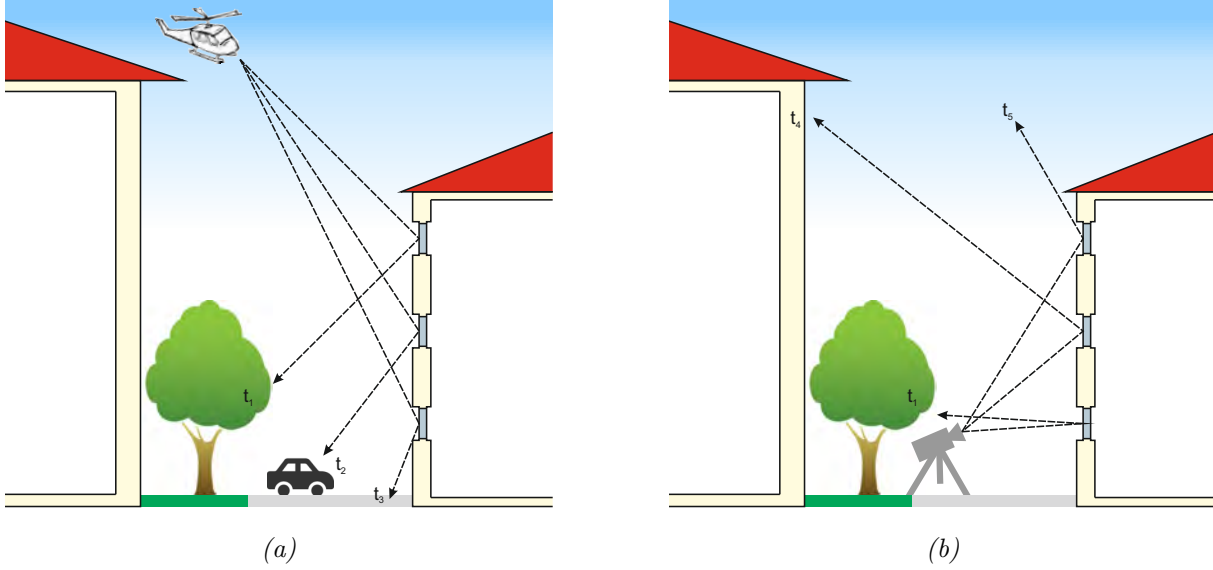
Figure 6.1: Reflections in windows. Depending on the viewing angle of the camera, different objects of the surroundings are reflected: a) airborne case; b) terrestrial case.

Although few 3D building models already include windows, most 3D datasets are created in Level of Detail (LoD) 2, in which the openings (windows and doors) are missing [Gröger et al., 2012]. Therefore, there is a need to develop methods to detect windows in thermal textures. In contrast to VIS images, not much training data is obtained in the TIR domain. Additionally, the resolution of the TIR images is typically lower than that of VIS images. Depending on the acquisition geometry, this can lead to difficulties in photographed façades or their parts, which have to be captured from a far distance and at a large angle - for example, in oblique airborne images or higher stories of the buildings from the street level. Fig. 6.2 presents examples of façades with windows taken from helicopter (about 400 [m] flight height) with an oblique TIR camera. Because windows are only a few pixels they are difficult to distinguish from the background. Fig. 6.3 presents some examples of window detected in TIR textures extracted from a terrestrial image sequence (Fig. 6.3 row A), and the same windows are detected in VIS images (Fig. 6.3 row B).

In the presented examples, the TIR images have lower resolutions and contrasts than VIS images, and the edges in TIR images are more blurred. Accordingly, it is difficult to model windows based on edges. Windows which appear very similar to each other in the VIS domain can appear very different in TIR images. For instance, the first two examples shown in Fig. 6.3 belong to the same façade. In VIS images (Fig. 6.3B-a and Fig. 6.3B-b), they appear very similar, while in TIR images, the same windows (Fig. 6.3A-a and Fig. 6.3A-b respectively) differ significantly from each other. Furthermore, in densely built urban areas the images taken from the street level usually capture a small field of view at a large angle; therefore, the window opening, which is orthogonal to the wall plane, can be seen on one side of the window. This is related to the fact that the windows do not lie in the plane of the façade; they are shifted inside. For example, in Fig. 6.3A-a and Fig. 6.3A-b, the image was taken from left, so the opening can be seen on the right. Moreover, the resolution for windows in upper stories (Fig. 6.3A-d) is significantly lower than that of windows in lower stories Fig. 6.3A-a, 6.3A-b, 6.3A-c). The problems with low resolution are even more remarkable for airborne TIR images. The expected size of a window in a thermal image can be in the range of a few pixels, depending on the used sensor and the distance. For example; a 4x2 [m^2] window on a façade captured with a camera at 45° angle and a distance
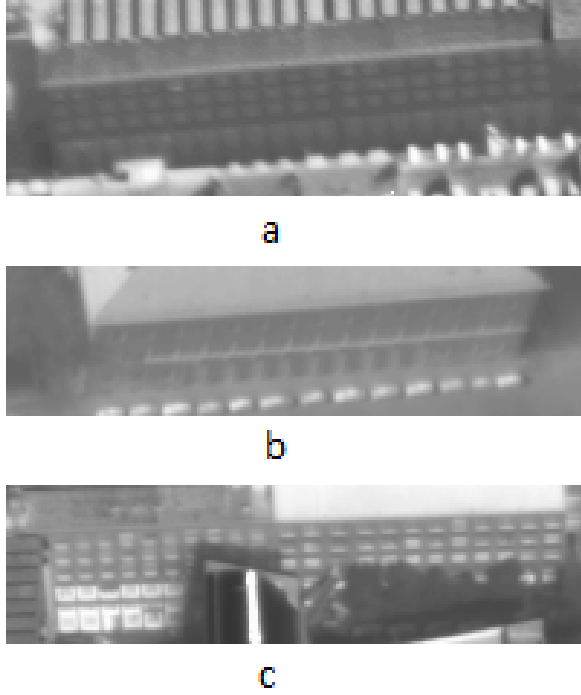
Figure 6.2: Examples of façades with windows in oblique airborne TIR images. Window size about: a) 8x6 [pix²]; b) 4x5 [pix²]; c) 6x5 [pix²]
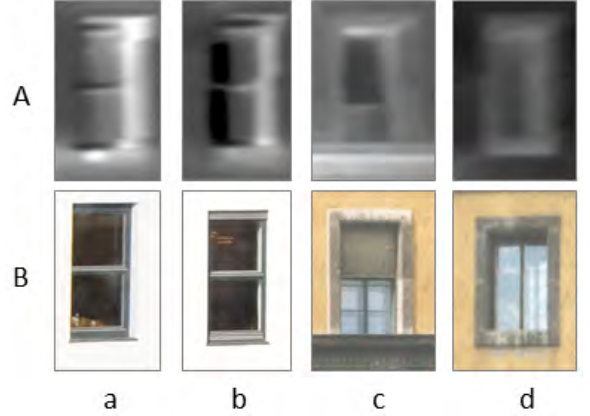

Figure 6.3: Examples of windows in terrestrial TIR textures and in terrestrial VIS textures. The row A shows windows from TIR textures created from terrestrial image sequences[Hoegner et al., 2007]. The row B shows the same windows (corresponding in columns) in the VIS image.

300 [m] with the focal length $f = 19$ [mm] and pixel size 17 [μm] will appear as an 5x10 [pix²] section.

6.2 Window and Façade Model

According to the properties of the TIR images, a window model is developed in this thesis and sketched in Fig. 6.4a. In this model, a window is simplified into three sections:

- a homogeneous background,

- a window frame,

- a reflecting pane (or sashes and smaller panes).

The window frame is usually built out of a different material than the material of the wall, and accordingly, very often has a different emissivity. Thus, compared with the background, the window frame appears with a different intensity. The area inside the frame consists of reflecting panes of glass and other elements. The window outline, which should be detected, is defined as the edge between the background and the window frame.

Smooth edges in the TIR image, the window outline is difficult to detect; however, the areas around the edges can be used for its detection. For this purpose, a masked correlation is used. The main idea of this method is that an edge is the border of two areas with different contrasts. If the intensity difference of both areas is not very large and the edge is smooth, the gradient magnitude is low. Accordingly, most of the standard contour detectors based on gradient imaging will fail to find this edge. However, if the shapes of these areas are known, they can be represented by a binary image and can be found in the image using cross correlation (Section 3.3.1) Based
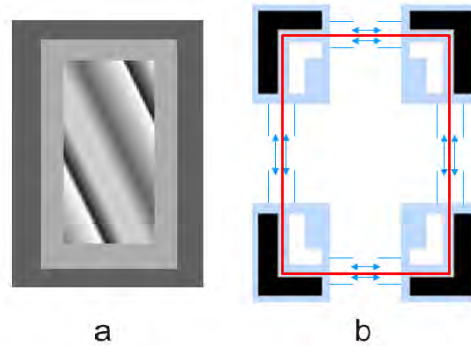
Figure 6.4: Window modeling: a) the window model (color coding: dark grey - homogeneous background, light grey - window frame, grey color gradient - glass); b) mask used for correlation (color coding: red - expected shape of the window, black & white binary mask, blue - don't-care-areas).

on the model sketched in Fig. 6.4a, the binary mask, which corresponds to the expected shape of the windows, is defined in this thesis. This mask is designed to search for edges separating the background from the window frame and is based on the rectangular shape with a predefined size. Fig. 6.4b shows such a binary mask for window detection. The mask represent an intensity change between the wall and the window frame, but only window corners are modeled. The mask consist of three areas:

- *on*-area (white) - an area with $1s$ in the binary mask,

- *off*-area (black) - an area with $0s$ in the binary mask,

- *don't-care*-area (blue) - an empty area in the binary mask not used for correlation.

It does not matter if the window frame is modeled using the *on*-area or the *off*-area because the absolute correlation value is calculated. The change of the intensity between the areas and the shape is essential; therefore, no information is needed about which material has a higher emissivity. Every corner is correlated separately, but the corners have to build a rectangle. This makes it possible to vary the size of the window. An additional advantage of correlating the corners independently is that you can deal with the occlusions related to the oblique view and with reflections caused by the specular surface of window panes. For example, Fig.6.3b shows a window observed from left bottom. The left side and the bottom of the window frame is not visible. Here, the glass pane, which reflected the sky, was imaged, rather than the window frame, as it would be it were an orthographic image. Hence, in this window, the left bottom corner shows a light to dark change from the wall to the window, and a dark to light change in the right upper corner. The edges between the background and the window frame are assumed to be blurred and are defined as *don't-care*-areas. The window panes are also defined as *don't-care*-areas and masked out for the correlation process. Correlation is calculated using eq. 3.11 proposed by Stilla [1993]. The mask is correlated within regions of interest (ROI), that means within regions where a window is expected.

A simplified façade model is used in this thesis. A façade consists of rectangular windows ordered in rows and columns. Windows are assumed to build a regular grid; however, missing windows are allowed. All windows in a row are assumed to be the same size, but this does not necessarily hold fine for columns. Fig. 6.5 presents some examples of possible window configurations in the façade.
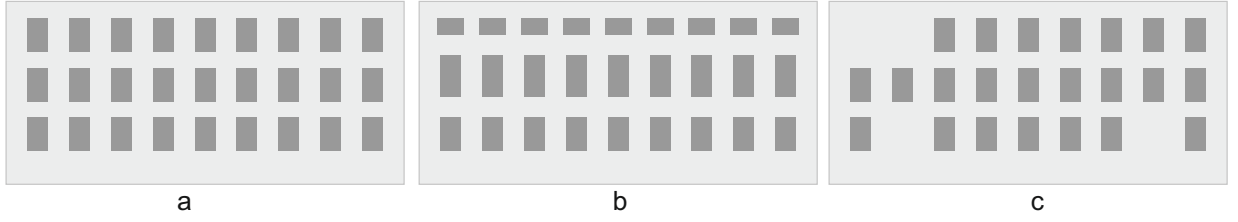
Figure 6.5: Possible façade structures: a) regular grid of windows of the same size; b) regular grid of windows of the same size in rows but different size in columns; c) grid of windows with some missing windows.

6.3 Regions of Interest

Regions of interest are parts of the façade where windows are expected. The correct definition of these regions is essential for starting the procedure for window detection. Some authors [Lee & Nevatia, 2004; Meixner & Leberl, 2010] recommend counting vertical and horizontal edge pixels to find the initial position for the windows. In TIR textures, however, the window edges are blurred and are more difficult to detect. Besides, other edges can often be detected in the façade image as a result of material differences in the façade or heat leakages; therefore, this work applies another technique to detect the ROI. For this purpose, dynamic thresholding is used, as described in Section 3.3.3. The approximate size of the window can be gathered from the 3D building model from the height and from the number of stories (e.g. from the cadaster) or from the scale of the TIR texture. Knowing this parameter, the windows are removed using the mean filter mask **G** of the size, approximately double size of the window. The difference per pixel is then calculated between the original and the smoothed image. This results in a different image, which is, then segmented using a local threshold (Section 3.3.3), as shown in eq. 3.14. Afterwords, bounding boxes around every detected segment are calculated. The gravity centers of the bounding boxes are used as starting points for the window search when using the masked correlation approach.

6.4 Lattice Reconstruction

Extracted gravity centers are used as preliminary candidates for windows and as starting points for the lattice reconstruction (Fig. 6.6). First, the candidates for windows are grouped into the stories. If the number of stories is known from the 3D building model, the grouping can be done using this information; however, this approach only works if all stories are the same height. If the number of stories is unknown, the grouping of the candidates for windows is carried out according to the approximate window size and to the distribution of the candidates for windows. All gravity centers are assigned height intervals, which are around half of the approximated window size. Next, a height histogram is created and the peaks of this histogram are detected. The final grouping of the candidates for windows correlates to the number of peaks in the height histogram, which is the expected number of stories.

The candidates for windows in each group are adjusted so that they share the same height because they are expected to build a row of windows of the same size arranged in one line. Afterwards, a search for corners is conducted around every possible window using the masked correlation introduced in Section 6.2. A window is detected if at least three corners are found. The detected windows in each group are then adjusted to the same height and the same size using the correlation coefficient as a weight. In all groups of detected windows, it is investigated

whether the windows are arranged in columns. If such a dependency is observed, the windows are also adjusted in columns, creating a regular lattice.

Finally, the lattice should be completed with the missing windows. This is accomplished by comparing the rows. In each row, the presence of windows in other rows at the same position are searched for. If detected, this gap becomes a candidate for a window. Using the masked correlation with the estimated window size for this row, this hypothesis is accepted or rejected. Additionally, the gaps between windows in rows are examined. If a window fits in the gap it is again candidate for window, which is verified using the correlation mask.
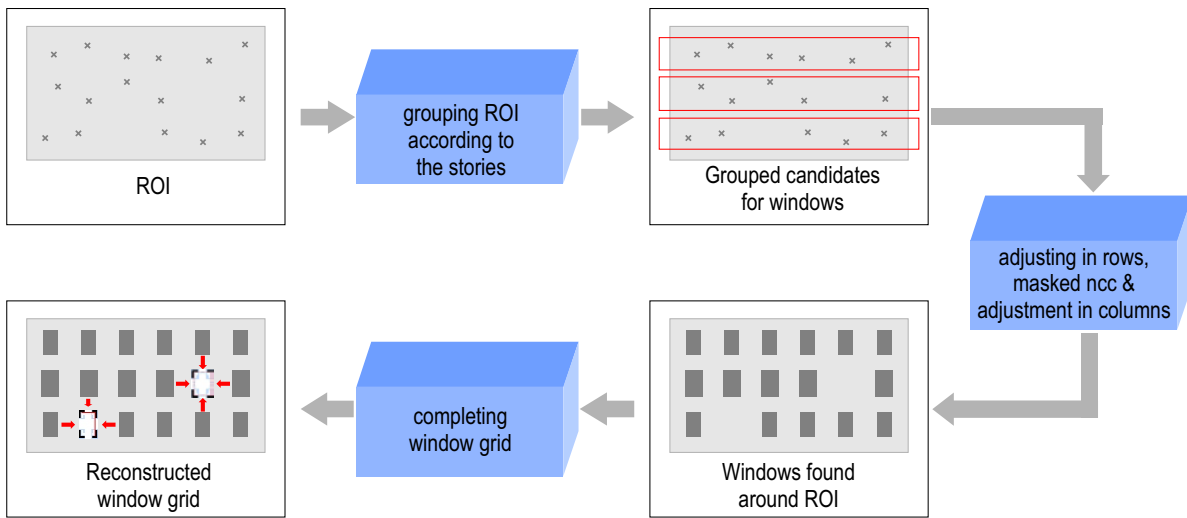
Figure 6.6: Reconstruction of the window grid

7 Experiments

The methodology presented in Chapter 4, Chapter 5, and Chapter 6 was implemented in MAT-LAB and tested on an experimental dataset. The data used for the experiments are described in Section 7.1, followed by the test end evaluation methods (Section 7.2).

7.1 Data Description

Flight campaign: In order to capture all of the façades and roofs in the test area, an appropriate flight campaign was designed and conducted. During this flight campaign, the helicopter flew four times above the test area, Techische Universität München, recording a sequence of oblique TIR images. As a result, four crossing stripes where captured over the test area. This flight campaign took place in autumn (October) 2006 in the early afternoon. The test area is located in the center of Munich, Germany and includes of the main campus of the university, including Old Pinakothek and the surrounding building complexes. This area is a densely built-up downtown area, where the buildings create ring-shaped structures with inner yards.

Infrared image sequences: The thermal images were taken with TIR camera AIM 640 QLW FLIR with a frame rate of 25 images per second, which was mounted on a platform carried by helicopter. The flying height was approximately 400 [m] above the ground level. The camera was forward looking with an oblique view of approximately $45°$. The size of the chip was 640x512 [pix^2]. The helicopter flew over the test area four times, recording four strips of IR image sequences. Each strip consists of almost 130 frames.

Navigation data: For direct geo-referencing, the data acquired by an Applanix POS AV 510 GPS/INS system with a 200 [Hz] frequency for INS and 1 [Hz] for GPS are used. GPS coordinates are used to correct the INS drift within the Kalman filter procedure [Grewal et al., 2007]. The recorded coordinates are referred to the center of the navigation device. The misalignment of the camera and GPS/INS coordinate systems is determined within an extended bundle adjustment and the ExtOri parameters are corrected and used for model projection.

System calibration: The camera calibration, boresight, and lever-arm parameters of the camera system used for the campaign were not known; therefore, the camera system was calibrated using the self calibration process presented in Kolecki et al. [2010]. The estimated boresight and lever-arm parameters were used to correct the navigation data, and the camera calibration parameters were used to undistort the images.

3D Building Model: The 3D building model of the test area was created using ArcScene - a commercial software for 3D building reconstruction from aerial images. Originally, the model was stored in 3D shape format and was converted to the CityGML format. During the conversion,

some basic semantics, such as surface type, were recovered. The transformed model is presented in Fig. 7.1.
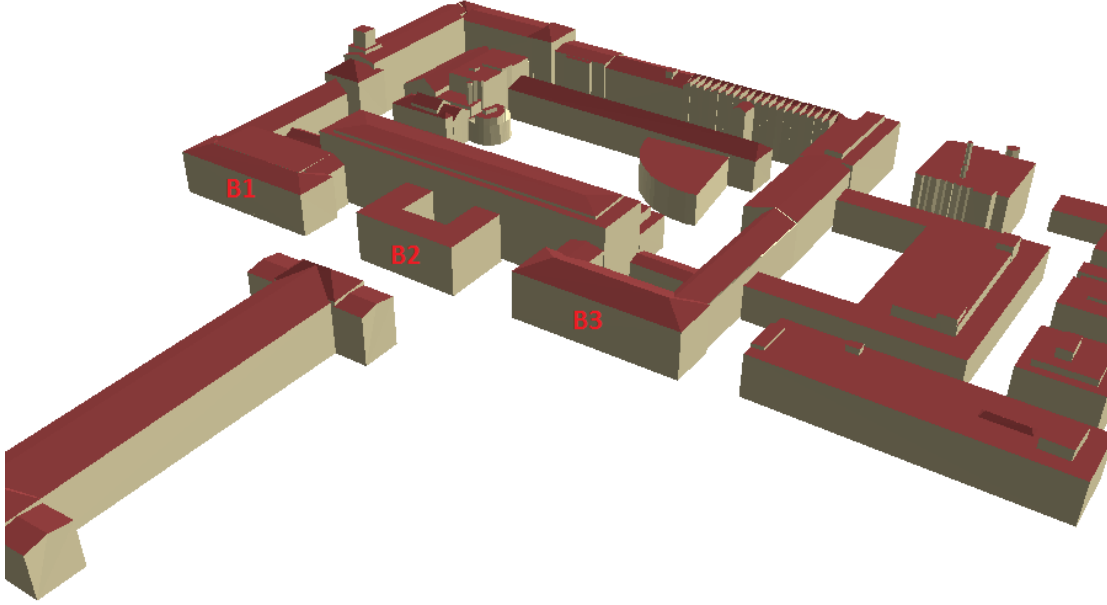

Figure 7.1: 3D building model of the test area "Techische Universität München"

7.2   Tests and Evaluation Method

Co-registration: Line segments detection was carried out with a commercial software MVTec HALCON using an operator for edge segment detection. This operator uses the Sobel filter for edge detection and the size of the Sobel mask as an input parameter [MVTec Software GmbH, 2014]. The size of the Sobel mask was set to SobelSize $= 9$. This operator also requires a minimum gradient magnitude: $1 \leq$ MinimumAmplitude ≤ 255. By setting the MinimumMagnitude, only pixels with a filter response larger than MinimumMagnitude are used as candidates for edge points. For these experiments, MinimumMagnitude was varied with MinimumMagnitude $= \{10, 30, 50\}$ and used as described in Section 4.1.1. Furthermore, the maximum allowed distance of an edge point to its approximating line was set to MaximumDistance $= 3$ [pix] and the minimum length of the line segments to MinimumLength $= 8$ [pix] in order to avoid very short lines.

Both methods for eliminating of outliers (RANSAC and accumulator) were tested and compared. These outlier detectors were evaluated visually. The evaluation method was based on counting true positive matches (TP), all possible correct positive matches (P) - ground truth, and false positive matches (FP). All possible correct negative matches N (ground truth) were calculated as $N = N_{all} - P$, where N_{all} is number of all extracted lines. True negative $TN = N - FP$ and false negative $FN = P - TP$ matches were also calculated. Next, true positive rate $TPR = TP/P$, false positive rate $FPR = FP/N$, and false discovery rate $FDR = FP/(FP + TP)$ were calculated.

Both methods for optimal pose estimations introduced in Section 4.1.4 were also implemented and tested. The Gauss-Markov model was implemented together with RANSAC, and the Gauss-Helmert model together with the accumulator.

Some experiments were conducted with selected buildings labeled as B1, B2 and B3 in Fig. 7.1

Window detection: Window detection was carried out in exemplary textures (Fig. 7.2) extracted from airborne images and, for comparison, from textures extracted in Hoegner et al. [2007] from terrestrial image sequences. The approximate size of the windows was derived from the façade size and from the texture size in pixels. The initial window size in the world coordinates was assumed to be around 2.5×2 [m^2]. For evaluation, ground truth windows were obtained manually, and the results were compared with the ground truth data. The correctness and completeness were then calculated as

$$correctness = \frac{TP}{FP + TP},$$ (7.1)

$$completeness = \frac{TP}{P},$$ (7.2)

where TP - correctly detected windows (true positive), P - all manually determined windows (ground truth), FP - falsely detected windows (false alarm), and $FP + TP$ - all detected windows. Due to the low resolution of TIR textures, blurred edges, and reflections, the position and size of the detected windows are not expected to be very accurate. For evaluation, a window was assumed to be detected if at least half of its surface was covered by the detection. The decision as to whether or not a window was detected was made visually.
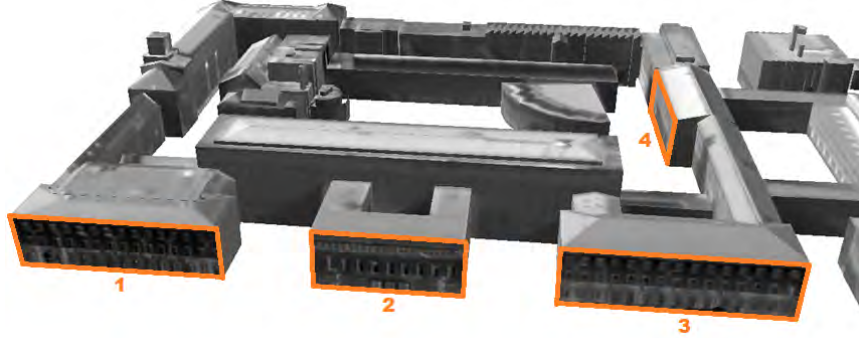
Figure 7.2: Test area for window detection. Four façades were selected for the experiment. For façades 1-3, terrestrial and airborne textures were available, while for façade 4, the airborne texture was available only in the yard.

8 Results

In this chapter, the experimental results using data described in Section 7.1 are presented. The results are ordered as follows: results on co-registration (Section 8.1), results on texture extraction (Section 8.2), and results on window detection (Section 8.3).

8.1 Results on Co-registration

Co-registration of the image sequences with the 3D building model (described in Section 7.1) was carried out using the method presented in Chapter 4.

Line segment extraction and weighting: First, line segments were extracted in all images and weights were assigned to them according to (4.1). The result of extracting and weighting line segments in an exemplary frame (Fig. 8.1a) is presented in Fig. 8.1b. In the presented example, three different settings for minimum edge strength ($g = \{10, 30, 50\}$) were used for the extraction. The bottom 20% of the lines with the lowest weights were precluded from the further processing.
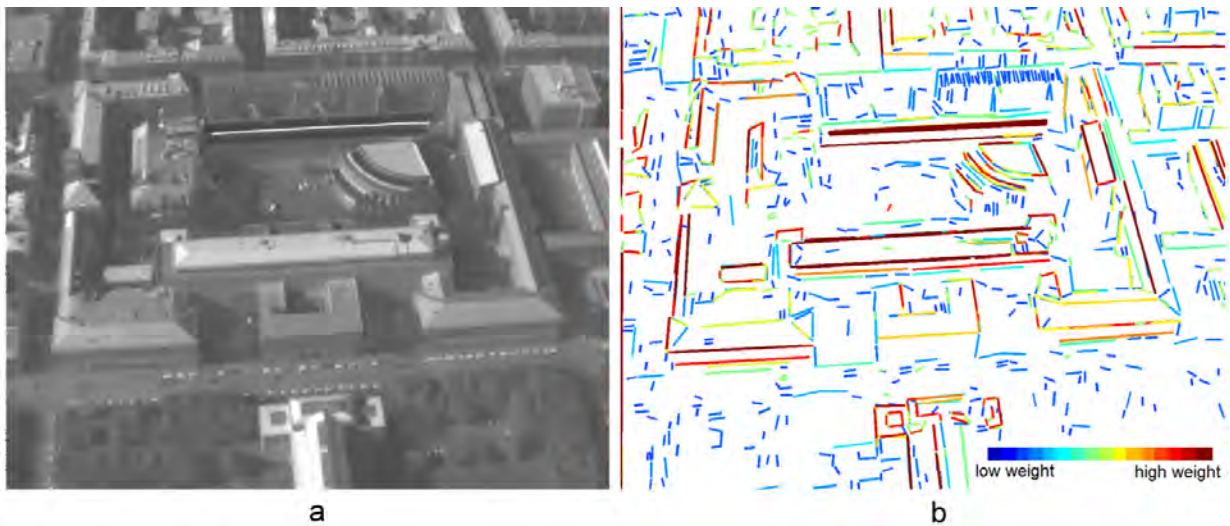
Figure 8.1: Results on line segment extraction and weighting: a) an exemplary TIR frame; b) weighting the line segments extracted in this frame. Three different minimum edge strengths were used for the extraction ($g = \{10, 30, 50\}$). Color coding from lowest to highest weight: blue, cyan, green, yellow, orange, and red.

Visibility check: Visibility check was applied as described in Section 4.1.2. Exemplary outputs with visible lines are presented in Fig. 8.2. Here, four frames were selected and each of these four frames were taken from different directions. The frame in Fig. 8.2a faces northward, frame in Fig. 8.2b faces southward, frame in Fig. 8.2c faces eastward and frame in Fig. 8.2d faces westwards.

(a)   *(b)*

(c)   *(d)*
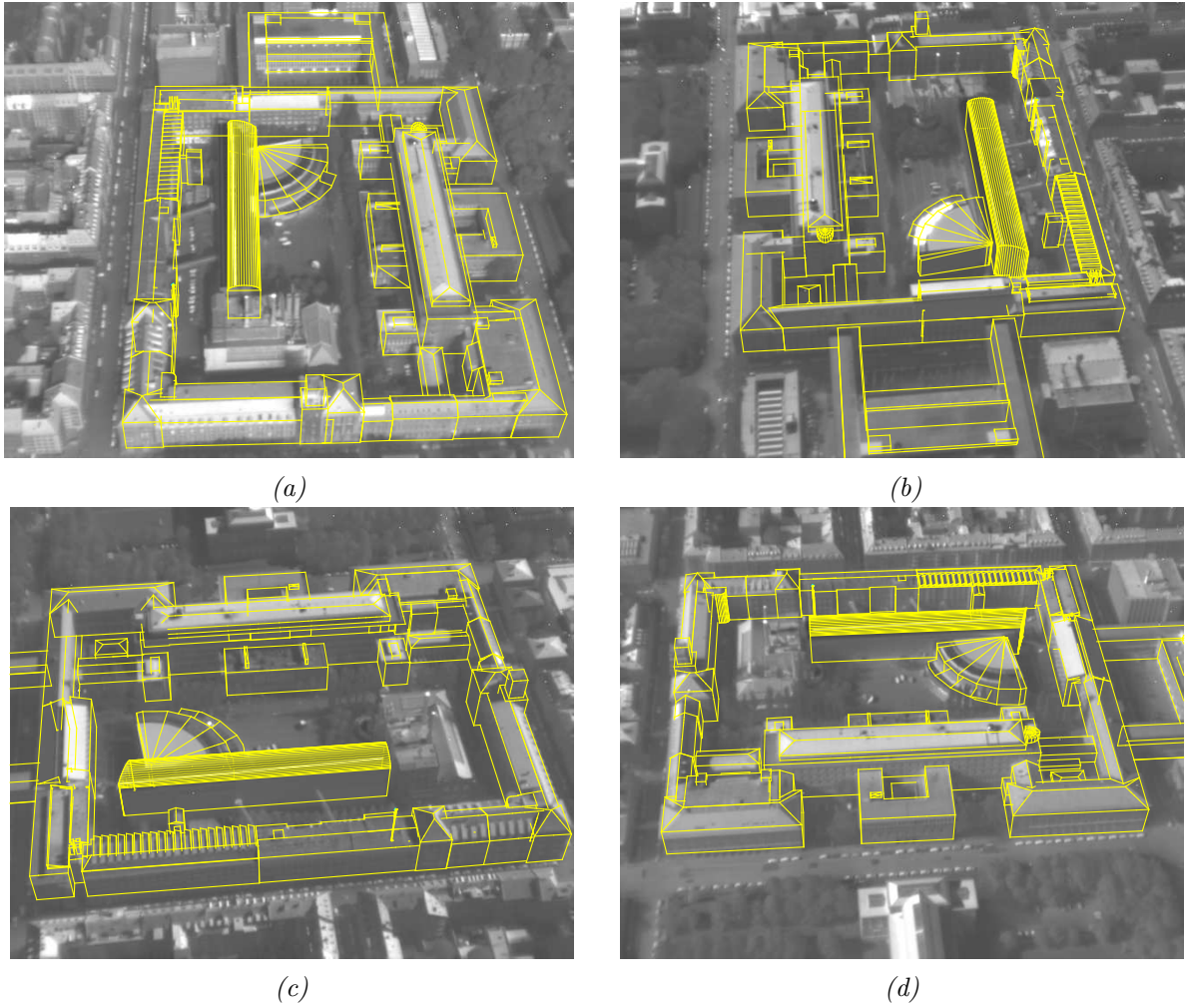
Figure 8.2: Results on visibility check for lines.

Model-to-image Matching: Model-to-image matching with optimal pose estimation was conducted as described in Section 4.1.4. For the evaluation, the subsets of building models, including buildings B1, B2 and B3 (see Fig. 7.1) were selected. Each frame was treated as an independent single frame. No correspondences between the frames were considered. The results for all steps of the matching procedure are illustrated in Fig. 8.3 using an exemplary frame. First, preliminary correspondences between the model edges and the image edges were searched in the surrounding of the model edge (Fig. 8.3a). Then, the preliminary correspondences were reduced using one of the outlier detectors (Fig. 8.3b). In this example, the correspondences were found using the accumulator approach. Finally, the selected correspondences were used for the optimal pose estimation. The adjustment was carried out in the projective space, and the model projection using estimated exterior orientation parameters is presented in Fig. 8.3c.

A similar example is presented in Fig. 8.4, showing a section of an image before adjustment (Fig. 8.4a) and after adjustment (Fig. 8.4b). For comparison, RANSAC was used to verify the correspondence in this example. Fig. 8.4a shows the results of the correspondence selection and uses the following coding: blue signals that the preliminary correspondences selected in the buffer approach where accepted by RANSAC, while magenta signals that they were rejected. Fig. 8.4b shows the same image section with the projected model before and after adjustment using red and green respectively. For the exemplary frame found in Fig. 8.4, the standard deviations for
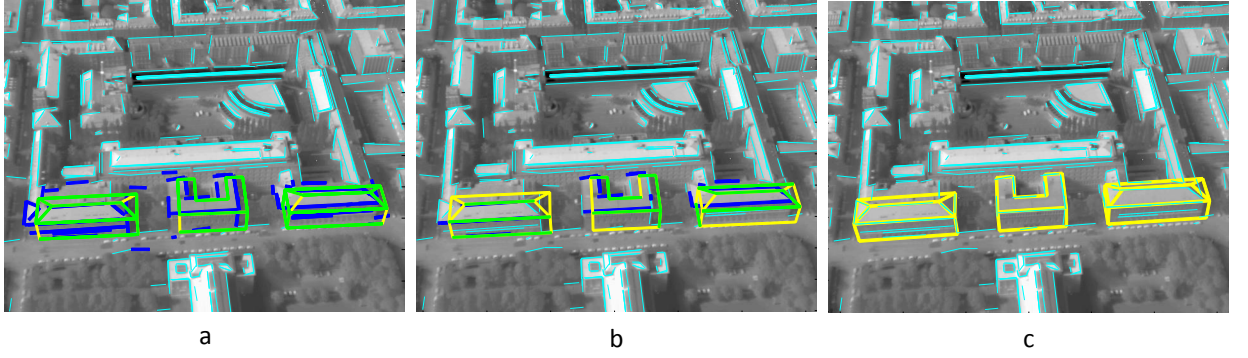
Figure 8.3: Exemplary result on model-to-image matching: a) line correspondences selected in the buffer approach (color coding: blue - image lines assigned to the model lines, cyan - image lines not assigned to the model lines, green - projected edges lines with corresponding image lines, yellow - projected model edges without corresponding image lines); b) line correspondences verified with accumulator (color coding: same as found in image a); c) extracted image lines (cyan) and model edges projected using estimated exterior orientation parameters (yellow).

the estimated exterior orientation parameters are $\sigma_X = 3.8$ [m], $\sigma_Y = 6.5$ [m], $\sigma_Z = 4.5$ [m], $\sigma_\omega = 0.60°$, $\sigma_\phi = 0.69°$, and $\sigma_\kappa = 0.39°$.
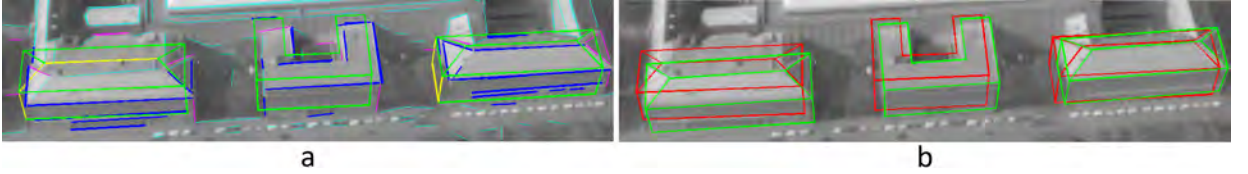
Figure 8.4: Section from an exemplary frame with extracted lines and a projected model: a) before matching (color coding: cyan - image lines without correspondences, magenta - image lines that were selected as correspondences by the buffer approach but refused by RANSAC, blue - image lines selected as correspondences and accepted by RANSAC, green - model lines with correspondences, yellow - model lines without correspondences); b) after matching (color coding: red - a 3D building model projected with initial exterior orientation parameters, green - a 3D building model projected with adjusted exterior orientation parameters).

Elimination of outliers: The elimination of outliers was tested using both the RANSAC approach and the accumulator approach. In order to compare the robustness of RANSAC and the accumulator, an additional test was conducted. Ten frames were selected and the algorithm for correspondence selection was carried out using (1) the accumulator and (2) RANSAC. For each of these ten frames, the true positive rate (TPR), the false positive rate (FPR) and the false discovery rate (FDR) were calculated, when using both (1) accumulator and (2) RANSAC. The mean of each rate, TPR, FPR, and FDR, for all investigated frames were then computed. The results of this evaluation and comparison of outlier detectors is presented in Tab. 8.1.

Tab. 8.1 shows that FPR was very low for both outlier detectors. This is due to the large number of all possible correct negative matches N. The assignment was carried out using only a portion of the image, because the test buildings (B1, B2 and B3) cover about 20% of the frame. N depends on all lines segments detected in the frame; therefore, FDR was also calculated, which is more representative in this case. In Tab. 8.1, can be seen that accumulator achieves better results: the TPR is higher, while FPR and FDR are lower.

Table 8.1: Evaluation and comparison of outlier detectors. N - all possible correct negative matches, P - all possible positive matches, TPR - true positive rate, FPR - false positive rate, FDR - false discovery rate

Frame	N	P	ACCUMULATOR			RANSAC		
			TPR [%]	FPR [%]	FDR [%]	TPR [%]	FPR [%]	FDR [%]
13200	724	34	76,47	0,55	13,33	52,94	0,69	21,74
13201	731	33	63,64	1,09	27,59	60,61	1,64	37,50
13202	738	34	76,47	0,54	13,33	64,71	0,68	18,52
13203	735	32	59,38	1,50	36,67	37,50	1,90	53,85
13204	745	32	53,13	1,74	43,33	59,38	1,07	29,63
13205	755	33	51,52	1,46	39,29	48,48	1,59	42,86
13206	773	34	88,24	0,26	6,25	52,94	0,91	28,00
13207	771	33	60,61	1,30	33,33	57,58	1,69	40,63
13208	762	35	28,57	1,31	50,00	60,00	1,18	30,00
13209	778	34	73,53	0,77	19,35	55,88	1,67	40,63
MEAN [%]			63,15	1,05	28,25	55,00	1,30	34,33

Optimal pose estimation: Optimal pose estimation was implemented and tested using the Gauss-Markov model in the Euclidean space and the Gauss-Helmert model in the projective space. In the approach with the functional model defined in the Euclidean space and with estimation formulated using Gauss-Markov model, special dealing with vertical lines is needed. Because covariance matrices for the line parameters of the vertical lines cannot be calculated using the error propagation approach, the covariance matrices for those lines are created based on error propagation for non-vertical line of the same length. Tests also showed several numerical instabilities in the adjustment when image lines were vertical in the image coordinate system.

To evaluate both methods and to investigate their sensitivity with respect to changes in the initial exterior orientation, a subset of buildings B1, B2, B3 and a sub-sequence of frames 100 frames (#13141 to #13240) were selected. The accuracy of the initial exterior orientation parameters were downgraded using white Gaussian noise. The noise was generated using normally distributed random numbers with mean $\mu = 0$ and standard deviation $\boldsymbol{\sigma} = [\sigma_{XYZ}, \sigma_{\omega\phi\kappa}]$, where $\sigma_{XYZ} = 1$ [m], $\sigma_{\omega\phi\kappa} = 0.1°$. Then the matching was carried out for every randomly downgraded set of exterior orientation parameters. Subsequently, the same test was successively repeated increasing the noise parameter vector to $3\boldsymbol{\sigma}$, $4\boldsymbol{\sigma}$, $5\boldsymbol{\sigma}$ and $7\boldsymbol{\sigma}$. The test was conducted using the Gauss-Markov model without any outlier detector (Tab. 8.2, row (1)), using RANSAC and Gaus-Markov model (Tab. 8.2, row (2)) and using the accumulator and the Gauss-Helmert model (Tab. 8.2, row (3)).

The size of the search space S_i and the angular threshold are calculated based on the expected displacement and rotation of the projected model lines, which are in turn calculated from the propagation of errors in the 3D building model and the exterior orientation parameters.

For example, while downgrading the initial exterior orientation by $4\boldsymbol{\sigma}$, the width parameter of the buffer $\Delta d = 3\sigma_d$ is in the range of about 40 [pix], and the angular threshold $\Delta\gamma = 3\sigma_\gamma$ is about 8°. The values are so high, because σ_d and σ_γ are calculated as a propagation of the uncertainties of the camera position and orientation, and the uncertainty of the 3D building model. This leads to many incorrect correspondences pre-selected using the buffer and the outlier rate $\epsilon \approx 85\%$. In case of downgrading with $7\boldsymbol{\sigma}$, the outlier rate increases to $\epsilon \approx 93\%$. The typical least squares adjustment method cannot cope with such a large number of outliers. Therefore, using RANSAC or the accumulator is necessary in this case. This algorithm is robust even if the camera is shifted

Table 8.2: Percentage of successfully matched samples with downgraded initial exterior orientation. $\boldsymbol{\sigma}$ *denotes the standard deviation used for the generation of normally distributed random numbers, with* $\boldsymbol{\sigma} = [\sigma_{XYZ}, \sigma_{\omega\phi\kappa}]^T$, *where* $\sigma_{XYZ} = 1$ *[m], and* $\sigma_{\omega\phi\kappa} = 0.1°$. *The Table shows (1) results based on the assignment of correspondences without outlier detection and the Gauss-Markov model adjustment; (2) results using RANSAC and the Gauss-Markov model adjustment; (3) results using accumulator approach and the Gauss-Helmert adjustment.*

	Successfully matched samples when downgrading the exterior orientation with normally distributed numbers using mean $\boldsymbol{\mu = 0}$ and standard deviation				
	$\boldsymbol{\sigma}$	$3\boldsymbol{\sigma}$	$4\boldsymbol{\sigma}$	$5\boldsymbol{\sigma}$	$7\boldsymbol{\sigma}$
(1)	98%	43%	20%	16%	0%
(2)	96%	68%	65%	61%	46%
(3)	96%	87%	82%	83%	75%

from the initial position by a few meters. For $7\boldsymbol{\sigma}$, the rate of successfully matched frames is still on a relatively high level, with 46% using RANSAC and Gauss-Markov model for the adjustment, and 75% using accumulator and Gauss-Helmert model.

In Fig. 8.5, some examples of successful matching after downgrading the initial exterior orientation parameters are shown. Various frames were selected to present the results. Each sub-figure shows initial parameters downgraded with normally distributed random numbers with another multiple of $\boldsymbol{\sigma}$ (the same as in Tab. 8.2).
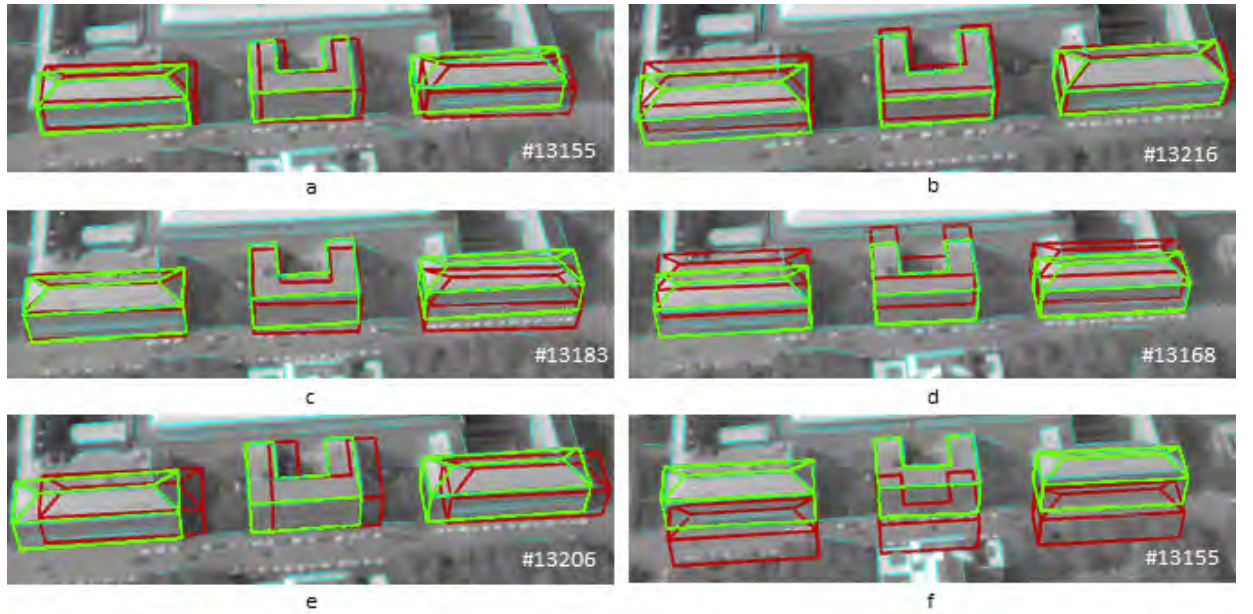
Figure 8.5: Examples for matching successful TIR images and the 3D building model projected with a) initial exterior orientation parameters; b) exterior orientation parameters downgraded by normally distributed numbers with $\boldsymbol{\mu}$ *and* $1\boldsymbol{\sigma}$; *c) with* $\boldsymbol{\mu}$ *and* $3\boldsymbol{\sigma}$; *d) with* $\boldsymbol{\mu}$ *and* $4\boldsymbol{\sigma}$; *e) with* $\boldsymbol{\mu}$ *and* $5\boldsymbol{\sigma}$.

In order to investigate the influence of the uncertainty on the matching results, tests with and without uncertainties were carried out. First, the method using Gauss-Markov model and RANSAC were evaluated. For this purpose, 82 frames were used and assessed visually, whether the matching result is better or worse when considering the uncertainties or the matching quality remains the same. This evaluation showed that in 48% cases, the matching was better when

considering the uncertainties. Only in 8% of the cases, the matching was worse. In other 44% frames, it was not possible to say which result was better. Three exemplary frames used for this test are presented in Fig. 8.6.
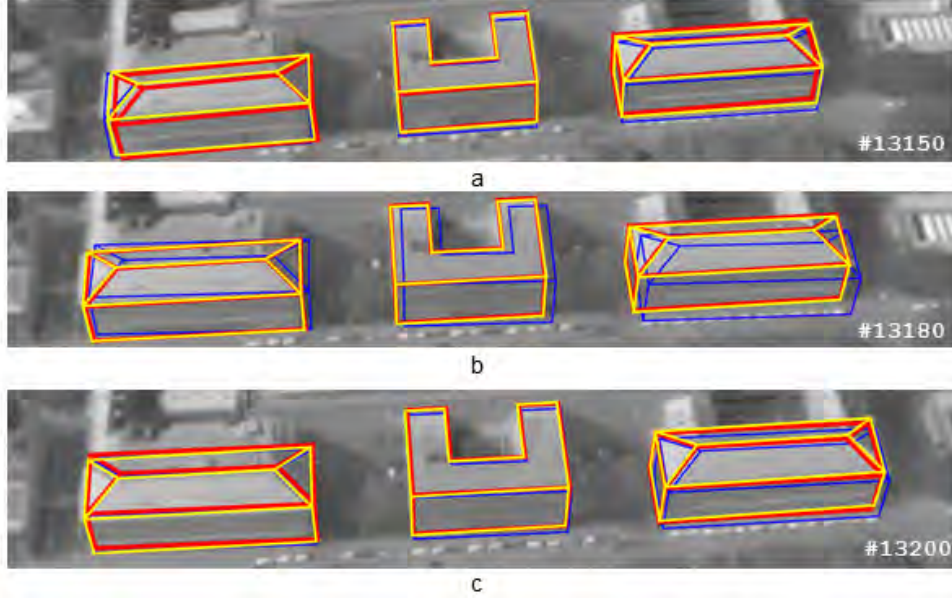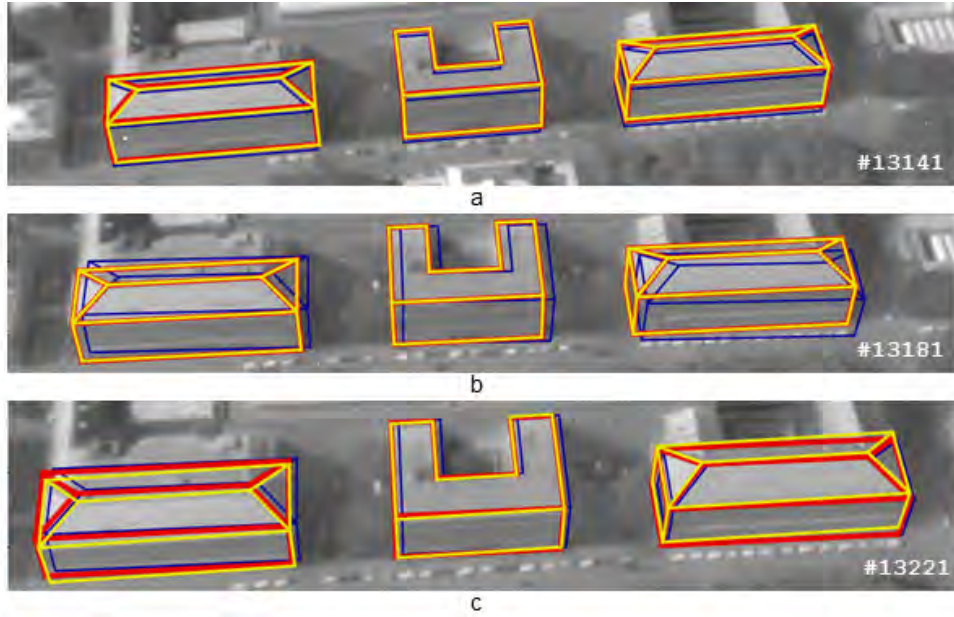

Figure 8.6: Influence of the uncertainty on the matching results in the Gauss-Markov model: a-c) three exemplary frames. Color coding: blue - initial projection of the model edges, red - projected model edges after adjustment without considering the uncertainties; yellow - projected model edges after adjustment with the uncertainties.

The same test was carried out with the estimation using the Gaus-Helmert model (Fig. 8.7). In this case, however, the influence of considering the uncertainties was not observed.

Because the method that uses the Gauss-Helmert model with accumulator as the outlier detector, performs better than the method using RANSAC and the Gauss-Markov model, further tests were carried out only using the Gauss-Helmert model and the accumulator.

Conditioning: In all presented experiments with homogeneous coordinates, the conditioning of the coordinates was conducted as described in Section 4.1.4. In order to show the importance of this conditioning, some test were also conducted omitting this step. For these tests, the subset of the building model, including buildings B1, B2, B3 and a sub-sequence (frames with ids #13141 to #13240), were used. For each frame, estimation was carried out twice, with and without conditioning. For both estimations, the same set of correspondences were used. Due to the method described in Section 4.1.4, conditioning is carried out if $\max_{hE} < f_{\min}$. For the selected subset of model points, belonging to buildings B1, B2, B3, the factor $\max_{hE}$ was computed. This calculation resulted in $\max_{hE} = 0.0977$. Factor $f_{\min}$ was set to 0.1. The results of this test are presented in Fig. 8.8.

Three frames from the beginning (Fig. 8.8A), middle (Fig. 8.8B), and end (Fig. 8.8C) of the sub-sequence were selected for visualization, but the results are comparable along entire sub-sequence. The first column of images (Fig. 8.8a) shows the results without conditioning, while the second column shows the results (Fig. 8.8b) with conditioning.

Fig. 8.9 shows estimated exterior orientation parameters and their 2σ confidence intervals. Estimated parameters calculated with conditioning are presented in blue, and their confidence

Figure 8.7: Influence of the uncertainty on the matching results in the Gauss-Helmert model: three exemplary frames. Color coding: blue - initial projection of the model edges, red - projected model edges after adjustment without considering the uncertainties; yellow - projected model edges after adjustment with the uncertainties

region in light blue, while estimated parameters calculated without conditioning are presented in red, and their confidence region in light red. For comparison, the initial parameters are also plotted (in cyan).

Robust estimation: Within the adjustment, an outlier detector was implemented as presented in Section 4.3. This detector is more sensitive and can detect additional outliers than those detected by RANSAC or the accumulator. However, this detector cannot deal with large number of outliers and gross errors.

The significance level for testing standardized corrections for the observations (Section 4.3) was set to $\alpha = 0.1$. Each observation is represented by 11 parameters, 3 related to the image line $\mathbf{l}_j = [a_j, b_j, c_j]^\mathsf{T}$, and 8 related to model corners $\mathbf{X}_{1i} = [U_{1i}, V_{1i}, W_{1i}, T_{1i}]^\mathsf{T}$ and $\mathbf{X}_{2i} = [U_{2i}, V_{2i}, W_{2i}, T_{2i}]^\mathsf{T}$. Each parameter is treated separately; for each observation, one or more parameters can be identified as outliers. Outliers in the image feature and outliers in the 3D building model can be distinguish. Because the inaccuracies in the line detection can also affect the errors in the parameters of the model points, these observations are not sufficient for outlier detection in the 3D model.

Tests showed that both errors coexist. Outliers in the line parameters were detected in less cases than outliers in model corner parameters. In most cases where outliers in the line parameters were detected, they coexisted with errors in the parameters of model points. The outliers in image line parameters were detected for 151 edges in 100 frames, but only in 21 cases they did not coexist with outliers in the parameters of model points.

Fig. 8.10 shows results on the outlier detection in an exemplary frame. Here, all three possibilities are presented, taking the outliers in the image line parameters, model point parameters or both at the same time. For outlier removal and outlier free adjustment, the third option was used. The results of the adjustment after outlier removal are presented in Fig. 8.10d.
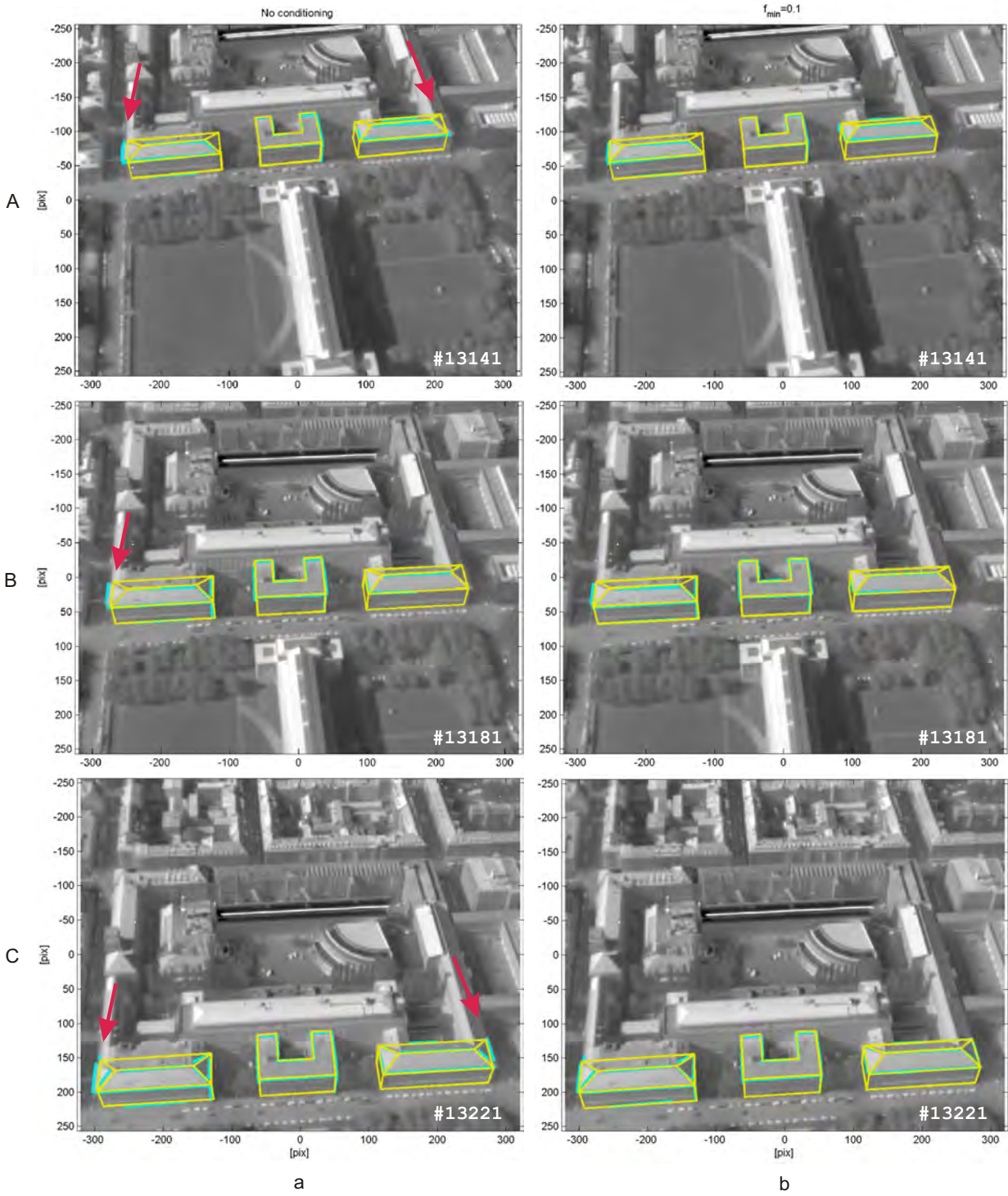
Figure 8.8: Influence of coordinate conditioning on pose calculation: a) estimation without conditioning; b) estimation using conditioned coordinates with factor $f_{min} = 0.1$; A) frame #13141 ; B) frame #13181; C) frame #13221 . Color coding: yellow - projected 3D building model after estimation, cyan - extracted image line segments which where used for the estimation
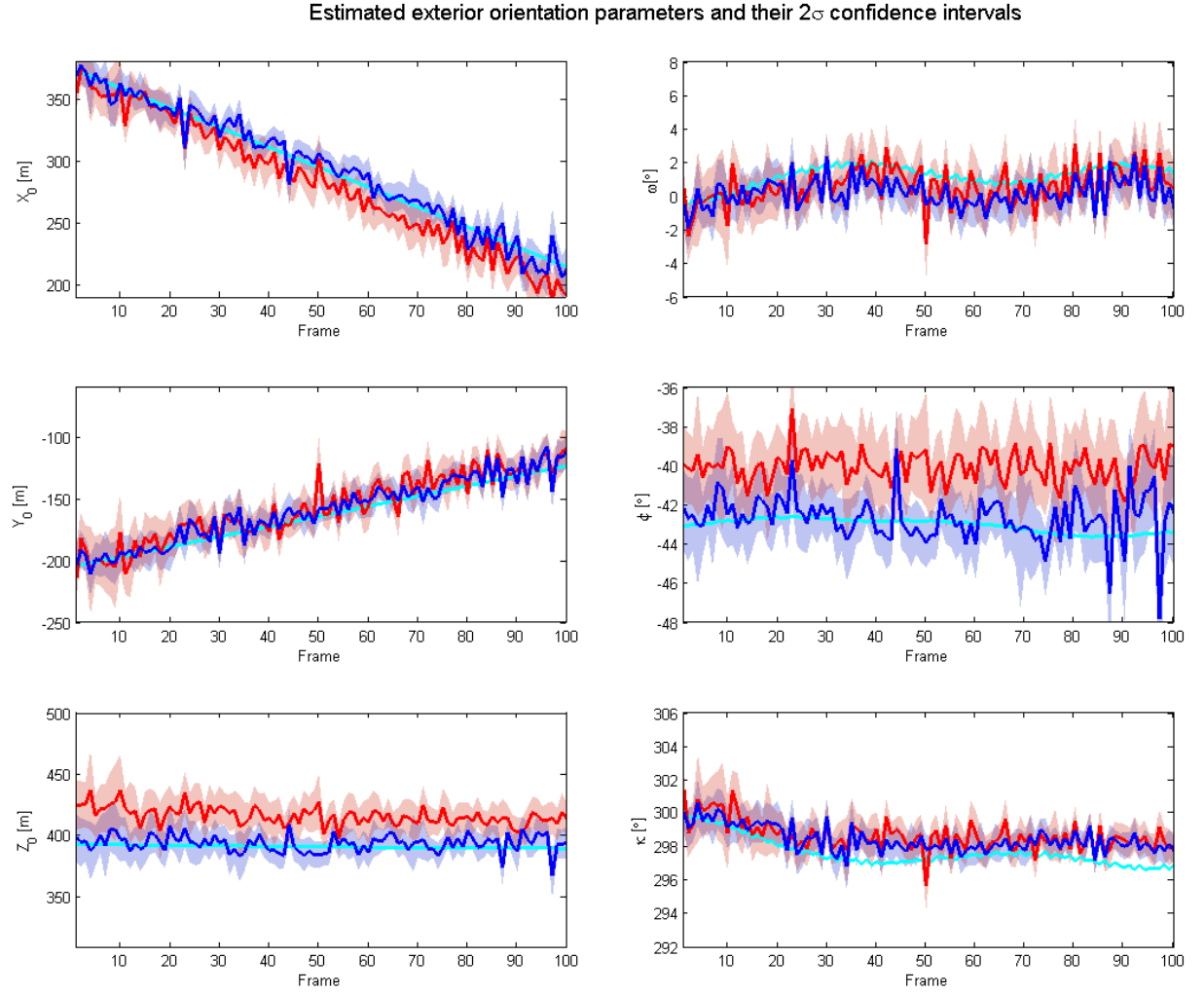
Estimated exterior orientation parameters and their 2σ confidence intervals

Figure 8.9: Estimated exterior orientation parameters and their 2σ confidence intervals. Color coding: cyan - initial parameters, blue - estimated parameters with conditioning, light blue - confidence region for estimated parameters with conditioning, red - estimated parameters without conditioning, light red - confidence region for estimated parameters without conditioning.

Fig. 8.11 shows the results of outlier detection considering all 11 parameters for each observation. The results are shown on one exemplary building and across a sub-sequence of six images.

Detected outliers are summarized for each edge along the sequence. If for an edge many outliers were counted, then one may suspect that this edge is badly modeled. However, not only the number of outliers counts, but also the relation to the overall correspondences found for this edge (Fig. 8.12) is important to be considered. This relation can be expressed by the number of outliers divided by the number of correspondences. Based on this measure, the outlieres in the models can be identified. Here, a threshold of 0.5 is selected to identify the outliers and threshold of 0.35 is used to identify possible outliers (Fig. 8.12b).

The outliers are visualized in Fig. 8.12c. Outliers are depicted in red and possible outliers in orange. In addition, Fig. 8.12c shows the edge IDs to enable the interpretation of the results found in Fig. 8.12a and Fig. 8.12b.
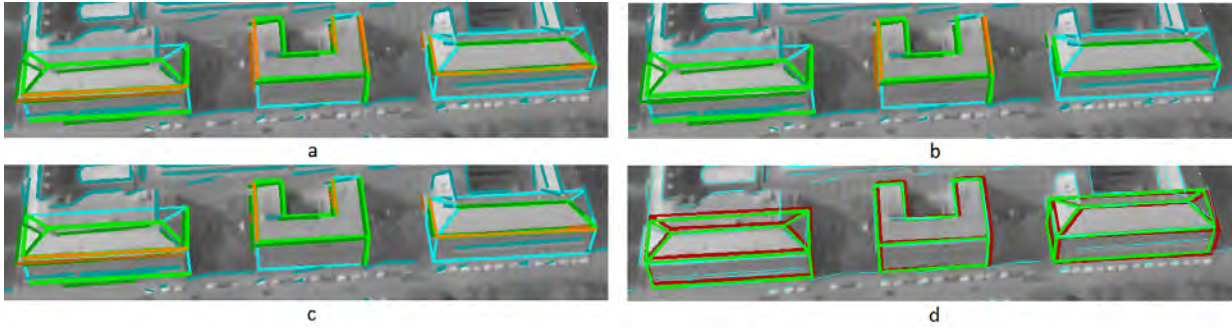
Figure 8.10: Results on outlier detection in the frame #13209: a) outliers detected based on the analysis of parameters of model points; b) outliers detected based on the analysis of parameters of image lines; c) outliers detected based on the analysis of all parameters; d) projected 3D building model before (red) and after (green) adjustment. Color coding for a-c: dark cyan - detected image line segments without correspondences, bright cyan - projected model edges without correspondences, dark green - detected image line segments with correspondences, bright green - projected model edges with correspondences, dark orange - detected image line segments with correspondences detected as outliers, bright green - projected model edges with correspondences detected as outliers.
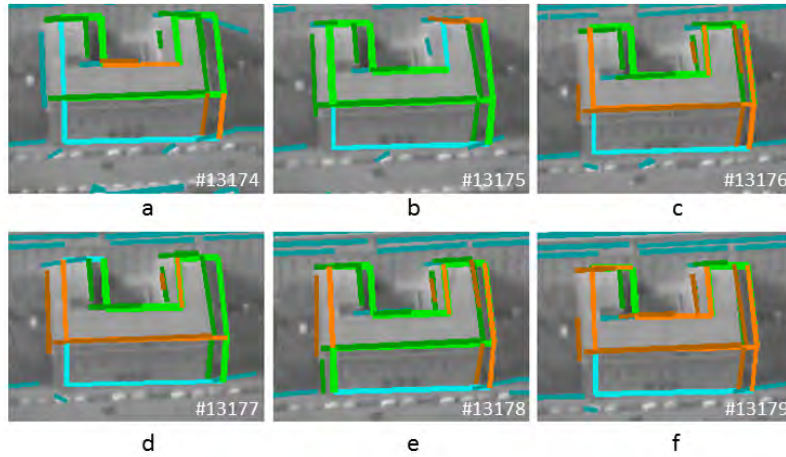
Figure 8.11: Outlier detection for one building across a sub-sequence of 6 images (#13174-#13179). Color coding: same as found in Fig. 8.10.

Tracking: To test the implemented tracking, pre-defined *key-frames* were used. The interval between the *key-frames* was set to 3, 5 and 7. The first frame was always defined as a *key-frame*. Exemplary results on tracking are presented in Fig. 8.13. In these figures, sections of the four following frames are shown. In the lower right corner of each image section, the ID of the corresponding frame was plotted. In the presented example, the interval between the *key-frames* was set to 3, hence frames #13141 (initial frame f_i with $i = 1$) and #13144 are *key-frames*, while frames #13142 and #13143 are normal frames.

Fig. 8.13a presents the projected model: in green - tracked model edges and in yellow - model edges, projected with estimated parameters. Fig. 8.13b shows the image line segments corresponding to the edges in the current frame (cyan) and the image line segments tracked as correspondences from the previous frame (blue).

Applying the presented tracking strategy, every projected model edge in frame f_i, where $i > 1$ can get two types of correspondences with the image edges:
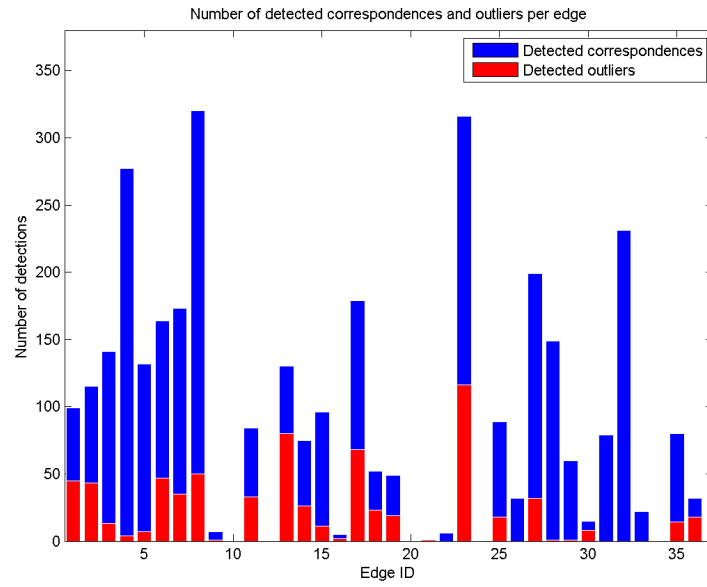
(a)
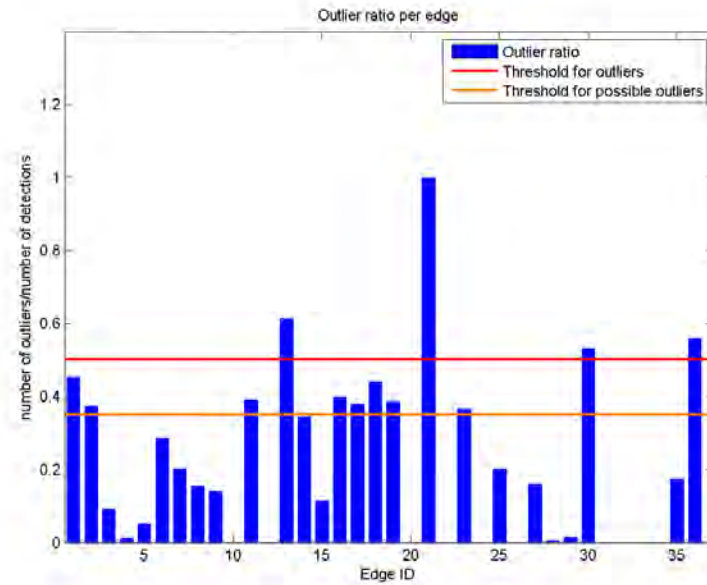
(b)
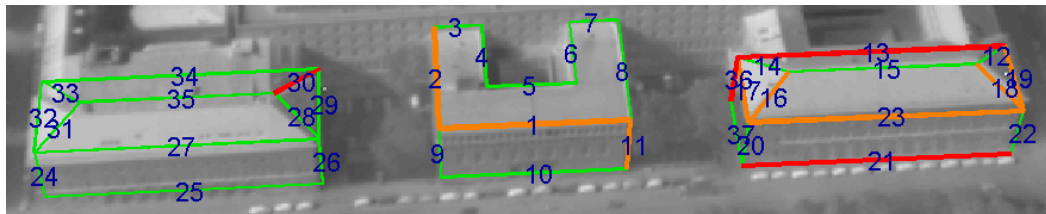
(c)

Figure 8.12: Outlier ratio: a) Number of detected correspondences (blue) and outliers (red) per edge; b) Outlier ratio per edge and threshold; c) edges detected as outliers (red), possible outliers (orange), and the edge IDs.
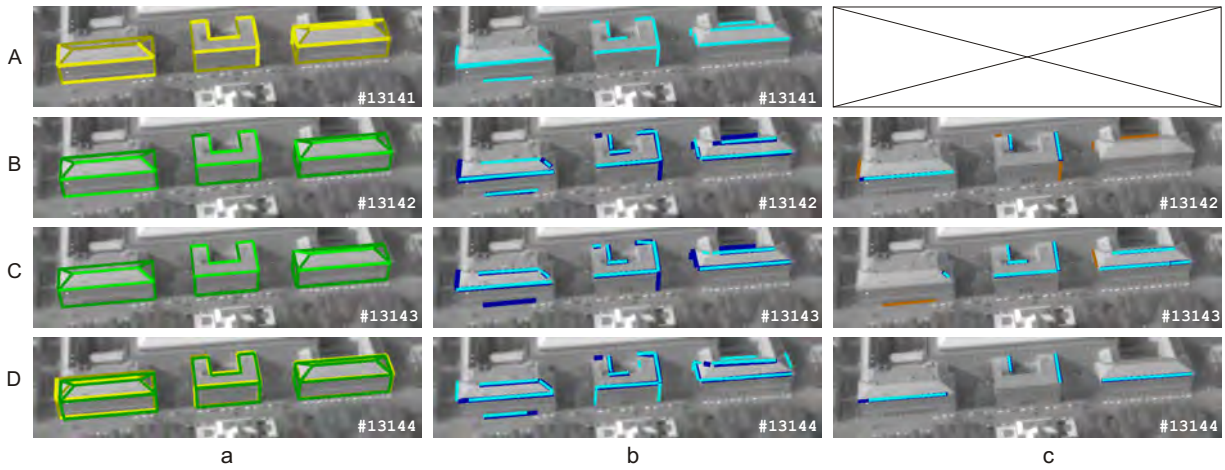
Figure 8.13: Image sections from a sequence of four images with two key-frames: a) with projected 3D building model (color coding: bright yellow - model lines with correspondences projected after parameter estimation, dark yellow - model lines without correspondences projected after parameter estimation, bright green - tracked model lines with correspondences, dark green - tracked model lines without correspondences); b) with plotted image line segments with correspondence (color coding: cyan - image line segments detected in current frame corresponding to a model edge, blue - image line segments tracked as correspondences from the previous frame); c) with verified correspondences (color coding: cyan - image line segments detected in current frame corresponding to a model edge, blue - verified virtual correspondences with correspondences in the current frame, dark orange - virtual correspondences which were added to the correspondence list and used for tracking in the next frame).

1. Assigned correspondences (with extracted edges)

2. Tracked correspondences (virtual, with tracked edges)

Virtual (tracked) correspondences can be helpful when not enough new correspondences are found in the current frame. However, they are not needed, if a new correspondence was found for a certain edge. Hence, in each frame, a verification of correspondences is carried out. It was tested whether or not there was a new correspondence which was equivalent to the tracked correspondence. This case occurs when the tracked image edge and the newly assigned image edge are equal. This is tested using three conditions:

- the middle points of the line segments are close to each other

- they are of similar length

- they are almost incident

The first two conditions are carried out by setting a threshold. For the third condition, statistical tests were implemented. If only a few correspondences were found in the current frame, then the missing correspondences are extended using the virtual correspondences. Hence, the virtual correspondences are also used for tracking in the next frame.

In order to assess the accuracy of the tracking, the model edges were also tracked into the *key-frames*. As a measure for this assessment, the distance between the tracked and projected model edges after estimation was used. For each corresponding pair of tracked and the projected model edges, the area between was calculated and divided by the length of the model edge. This

Table 8.3: Evaluation of tracking with adjustment only in key-frames. The quality is expressed as the average distance between the tracked and projected model edges. Here, analysis of this value per frame is presented.

Interval between the *key-frames*	3 frames	5 frames	7 frames
Average distance	1.6 [pix]	2.4 [pix]	3.1 [pix]
Maximum distance	2.2 [pix]	3.1 [pix]	4.0 [pix]
Minimum distance	0.9 [pix]	1.4 [pix]	0.8 [pix]

Table 8.4: Evaluation of tracking with adjustment in every frame. The quality is expressed as the average distance between the tracked and the projected model edges. Here analysis of this value per frame.

Interval between the *key-frames*	3 frames	5 frames	7 frames	9 frames	11 frames	20 frames
Average distance	1.3 [pix]	1.6 [pix]	1.0 [pix]	1.2 [pix]	1.3 [pix]	1.0 [pix]
Maximum distance	2.2 [pix]	2.7 [pix]	1.6 [pix]	1.6 [pix]	2.4 [pix]	1.5 [pix]
Minimum distance	0.6 [pix]	0.6 [pix]	0.6 [pix]	0.9 [pix]	0.7 [pix]	0.7 [pix]

value was considered to be the average distance between those two edges. This distance was summed up and averaged among the whole frame and then stored as the quality value per frame. Tab. 8.3 shows an analysis of these values stored per frame, dependent on the pre-defined interval between the *key-frames*. The interval between the *key-frames* was set to 3, 5 and 7 frames. In the first row of Tab. 8.3, the average distance between the tracked and the projected model edges after adjustment are presented. This value can be interpreted as an average shift, which would be made by extracting a texture in a *key-frame* using the tracked model.

The same test was carried out with adjustment in every frame. The results of this experiment are shown in Tab. 8.4. Here, the tests were carried out using *key-frame* interval 3, 5 and 7 frames, as well as *key-frame* interval 11 and 20 frames.

The computation time was also investigated[*]. The search for correspondences using the accumulator took about 16 [sec] per frame. The adjustment procedure, including the outlier detection within the adjustment, took about 0.6 [sec] per frame. Line tracking took about 0.1 [sec] per frame. Adjustment and tracking are carried out in every frame, but they search for correspondences only in *key-frames*. In the current implementation, the computation time is about 17 times higher in *key-frames* than in normal frames.

8.2 Results on Texturing

Best texture selection: Texture selection was carried out based on the quality measure presented in Section 5.6.1. The coefficients a_1, a_2 and a_3 were set to $a_1 = a_2 = 1$ and $a_3 = 2$. The results of the texture selection are presented in Fig. 8.14, whereas Fig. 8.14a shows the quality of the selected textures and 8.14b shows the viewing direction for the frame from which the texture was selected.

Local Matching: Results on local matching are presented in Fig. 8.15. In order to refine the fit between the 3D building model and the image, the model edges were matched locally in the

---

[*]using non-optimized MATLAB implementation

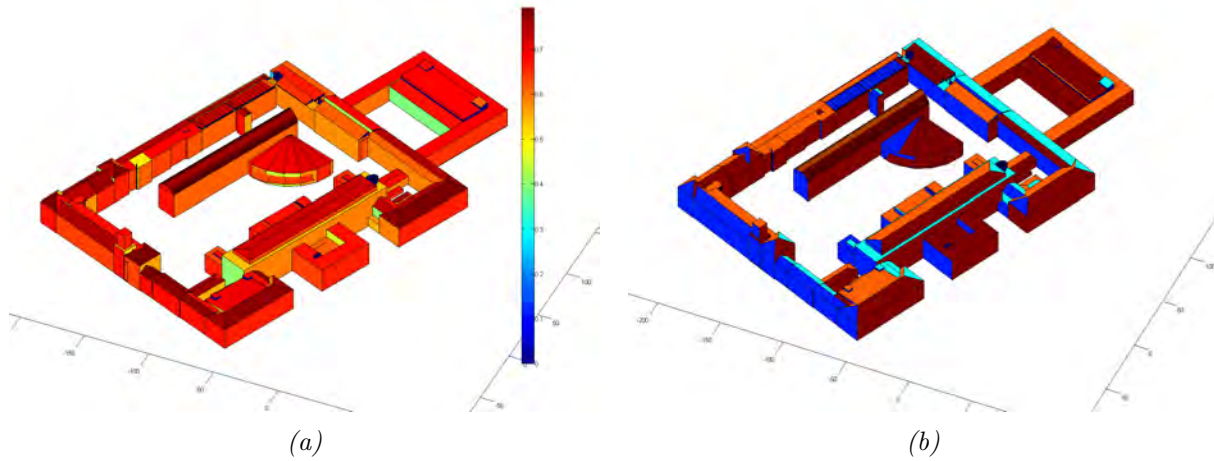*(a)*   *(b)*

Figure 8.14: Results on best texture selection: a) quality of selected textures; b) faces colored due to the flight direction of the selected best texture (color coding: dark blue - from south to north, cyan - from north to south, orange - from west to east, red - from east to west).

image. In doing so, small roof overlaps could be corrected, making the texture mapping more precise (see Fig. 8.15). The main advantage of this method is that the objects seen in the textures can be precisely geo-referenced in the wall coordinate system. The wall coordinate system can be transformed into the world coordinate system; however, for objects observable in thermal images, such as heat leakages, it is more important to localize them relatively to the building elements and not in an independent 3D world coordinate system.
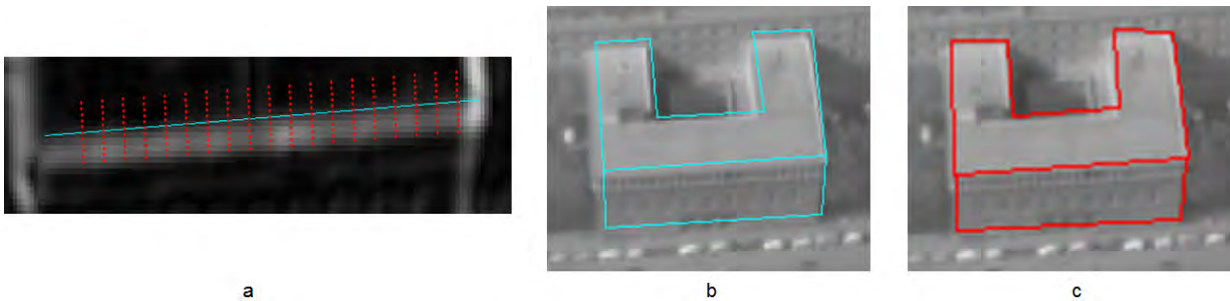
Figure 8.15: Local matching: a) adjustment for an exemplary edge; b) initial position of the projected 3D building model; c) position after local matching.

Texture extraction: For each face of the model, a TIR texture was created based on the results on best texture selection and local matching. Those textures were mapped on the 3D geometry in *CityGML* format and displayed using *Autodesk LandXplorer* (Fig. 8.16).

Matching quality: To evaluate the matching quality, the reference polygons were drawn manually in 70 frames. These reference polygons were used to calculate the quality measure ν from (5.18). First, the quality measure was computed for the initial model projection using the exterior orientation parameters after system calibration [Kolecki et al., 2010]. Then, the quality measure ν was calculated for projection using the exterior orientation parameters corrected by matching. These results are presented in Tab. 8.5. Next the matching quality was calculated after estimation with and without taking uncertainties int account. This was done for the estimation (1) in Eu-
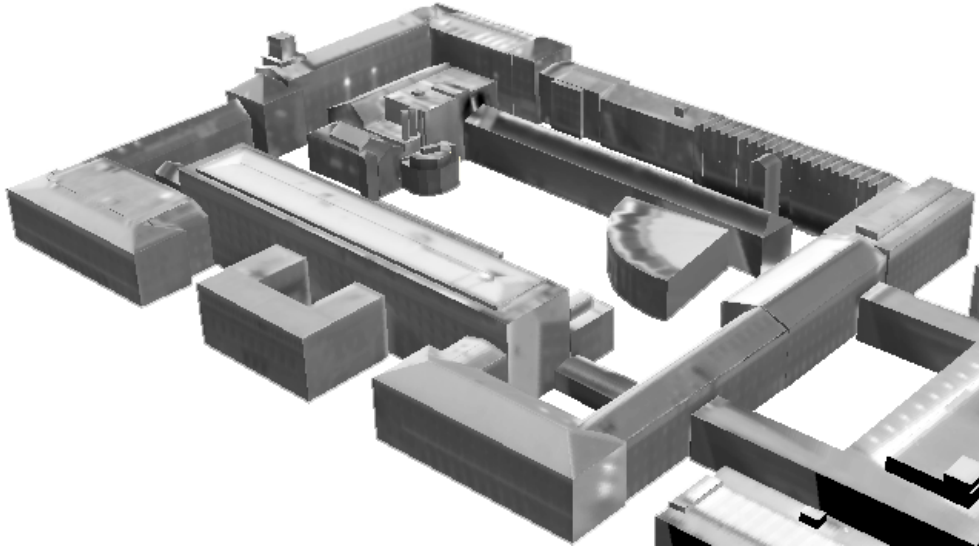
Figure 8.16: 3D building model with TIR textures

Table 8.5: Quality measure ν calculated after system calibration and after matching in every frame

ν [pix] after system calibration	ν [pix] after matching in every frame
4.74	1.48

Table 8.6: Quality measure ν calculated after estimation with and without uncertainties: (1) in Euclidean space - using Gauss-Markov model; (2) in projective space - using Gauss-Helmert model.

	ν [pix] for estimation with uncertainties	ν [pix] for estimation without uncertainties
(1)	2.95	4.33
(2)	1.48	1.73

clidean space using Gauss-Markov model, as well as, based on the same correspondences selected by accumulator, (2) in projective space using Gauss-Helmert model. The result of this experiment is presented in Tab. 8.6. Finally, the quality measure ν was also calculated after tracking. These results are presented in Tab. 8.7. This experiment was carried out for varying *key-frame* intervals.

Table 8.7: Quality measure ν calculated after tracking with varying key-frame interval

Interval between the *key-frames*	3 frames	5 frames	7 frames	9 frames	11 frames	20 frames	25 frames	30 frames
ν [pix]	1.55	1.64	1.73	1.64	1.67	1.71	1.71	1.73

Table 8.8: Extracted candidates for windows in terrestrial and airborne textures for four façades

	Number of detected gravity centers	
Texture	airborne	terrestrial
Façade 1	45	22
Façade 2	36	24
Façade 3	39	18
Façade 4	41	–

8.3   Results on Window Detection

This section presents results on automatic window detection in TIR textures. Fig. 8.17 depicts extraction of regions of interest (ROI) using a dynamic threshold. A terrestrial TIR texture (top left) was segmented using a dynamic threshold and segmented regions were smoothed using morphological operations (top right). Bounding boxes around the regions were created (bottom left) and the gravity centers of these bounding boxes, which are preliminary candidates for windows, were calculated (bottom right). Tab. 8.8 shows the number of detected candidates for windows in tested terrestrial and airborne TIR textures.
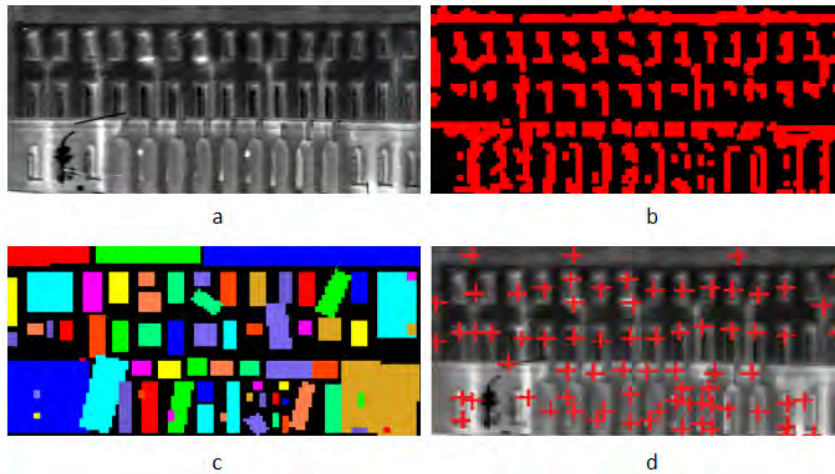
Figure 8.17: Candidates for Windows detected in a terrestrial texture: a) a TIR texture extracted from terrestrial data; b) regions of interest extracted from the TIR texture using local dynamic threshold and morphological operations; c) bounding boxes around the regions of interests; d) gravity centers of the bounding boxes.

Fig. 8.18 presents results on the lattice reconstruction for four exemplary façades. For Façade 4, only the airborne texture was available because the vehicle with the mounted TIR camera was not able to enter the courtyard.

In reality, Façades 1-3 feature four rows of windows, but the last row (fourth story) consists of significantly smaller windows than the other rows. In all textures of these façades a maximum of three rows of windows were detected. Evaluating the textures visually did not allow the evaluator to realize the existence of the fourth floor and the fourth row of windows; therefore, the fourth row of windows was not taken into consideration while evaluating the results. Façade 4 consists of five rows of windows and all of the rows were detected. Tab. 8.9 shows the evaluation of the detection algorithm using correctness and completeness as quality measures. For terrestrial textures, significantly higher correctness and completeness were achieved.
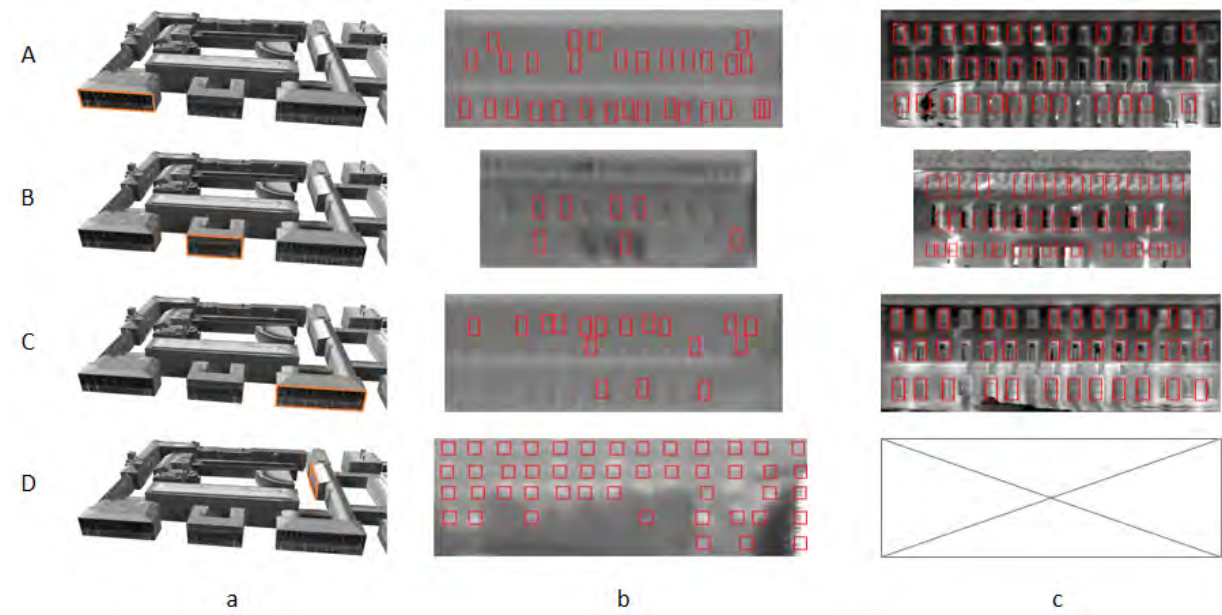
Figure 8.18: Windows extracted in airborne and in terrestrial textures of four façades: a) location of the façade marked in the textured 3D building models; b) windows extracted in the airborne textures; c) windows extracted in the terrestrial textures. Four different façades were used for the experiment: A) Façade 1; B) Façade 2; C) Façade 3; D) Façade 4.

Table 8.9: Evaluation of the window detection algorithm

	Façade 1		Façade 2		Façade 3		Façade 4		**Total**	
	airb.	terr.	airb.	terr.	airb.	terr.	airb.	terr.	**airb.**	**terr.**
correctness	30%	97%	57%	47%	24%	83%	100%	–	**63%**	**73%**
completeness	15%	73%	6%	57%	7%	67%	72%	–	**26%**	**66%**

Tab. 8.10 presents the size of windows detected in terrestrial textures. Detection was carried out using an initial window size of 2.50×2.00 [m]. For comparison, the size of the windows were measured manually and presented in brackets. Access to some windows was restricted, so some ground truth data is missing.

The detected windows have been embedded in the 3D building model stored in *CityGML*. For better visual perception, the windows were indented and colored in blue. It should be mentioned that the indention was not the result of any measurement, but was done only for a more realistic visualization. The 3D building model was displayed with the software *LandXplorer*. A screenshot from this software is presented in Fig. 8.19.

Table 8.10: *Size in meters of detected windows in terrestrial TIR textures. Presented values are expressed in [m]. When available, the manually measured ground truth size is presented in brackets.*

Story	Façade 1		Façade 2		Façade 3	
	height	width	height	width	height	width
1st	2.59 (——)	2.62 (——)	2.08 (1.25)	1.34 (1.25)	3.10 (——)	2.24 (——)
2nd	2.92 (2.85)	2.09 (1.50)	3.00 (3.50)	1.90 (1.65)	2.87 (2.85)	2.50 (1.50)
3rd	2.79 (2.15)	2.23 (1.50)	3.62 (——)	2.01 (——)	3.18 (2.15)	1.71 (1.50)
4th	–	–	–	–	–	–
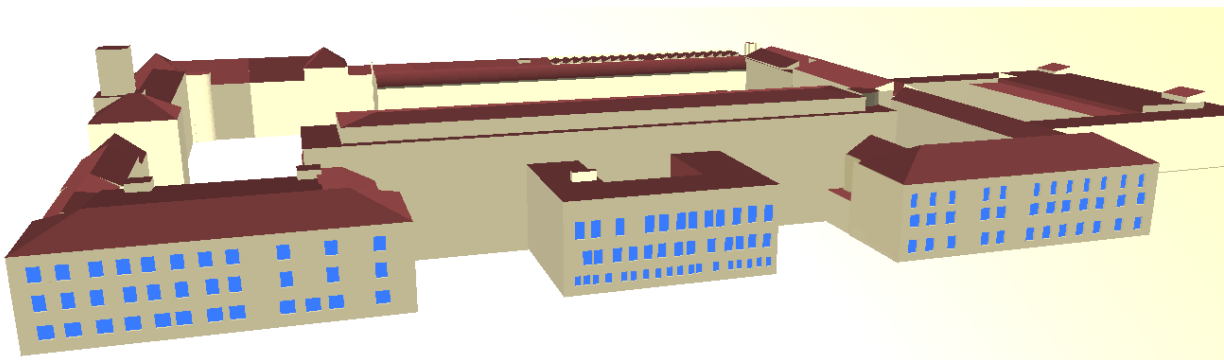
Figure 8.19: *Extracted windows embedded in a 3D building model displayed by LandXplorer.*

9 Discussion

In this chapter, the results presented in the previous Chapter 8 are discussed and evaluated. This chapter is organized in sections as follows: Co-registration (Section 9.1), Texturing (Section 9.2) and Detection of Windows (Section 9.3).

9.1 Discussion on Co-registration

Line segment extraction and weighting: Line based model-to-image matching has a high potential for co-registering building models with oblique airborne images. Edges are the most representative features for building structures and can be easily detected in the image using standard image processing algorithms. However, the standard line detector based on the Canny edge detector implemented in the HALCON Software, which was used in the experiments, did not provide very accurate results. This is due to the representation of the line segments by the coordinates of the end points in pixel accuracy. Sub-pixel accuracy and line extraction adjusted to the properties of TIR images could improve the results of the co-registration. The first attempt to improve line segment extraction implemented in this thesis involved extracting using different input parameters (edge strength), weighting and selecting the best lines. Improving the geometric accuracy in future studies, however, could increase the accuracy of the adjustment.

Visibility check: The method implemented for visibility check made it possible to find the visible model's edges and to maintain them as vector data. However, some of these problems remained after the implementation. Some of them are presented in Fig. 9.1. Fig. 9.1a shows an overview of the scene and the projected line segments that have been identified as visible by the implemented algorithm. The first problem, which can be observed in this scene, is that some of the visible lines are very close to each other (Fig. 9.1b and c). In Fig. 9.1b, this is caused by one roof with a repetitive structure and four chimneys. In this case, all the edges should have corresponding line segments in the image; however, their correct assignment can be difficult. In Fig. 9.1c, the model's edges are densely projected because of how the roof was modeled. In reality, it is an arched roof with a smooth surface and no visible edges. In the model, the same roof was modeled using long narrow polygons, which caused the model's edges to be very dense. In this case, these edges are not expected to have corresponding edges in the image. Even if some edges could be found in this area (e.g. due to sun reflections in the metal roof), it would not provide the correct assignment. Hence, in both cases, these edges should not be taken into account for the correspondence search. This should be the subject of a future study.

Fig. 9.1d shows another problem: due to numerical errors in the implementation of the visibility algorithm, some of the visible lines were identified as occluded (red arrow in Fig. 9.1d). In the presented example, some edges of a lower building, which overlaps with the yard of an another building, are missing. The problem of unmodeled building parts can also be observed in this model (red arrow in Fig. 9.1a). This problem will be discussed in Section 9.2.
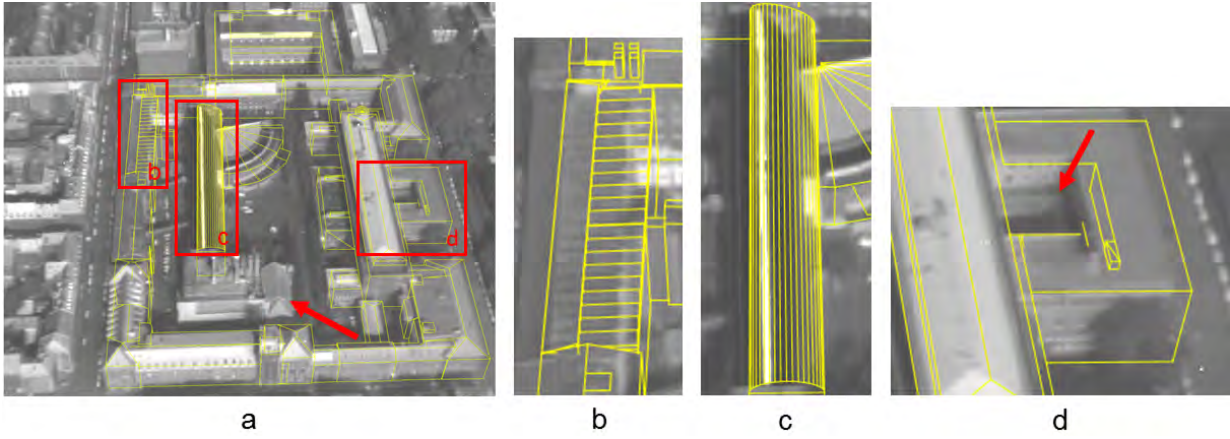
Figure 9.1: Difficulties in visibility check for mode edges: a) overview of the scene with projected model lines; b) roof with repetitive structures; c) arched roof; d) a zoom-in image section showing incorrectly identified line visibility

Model-to-image Matching: Both proposed methods for model-to-image matching delivered good results in terms of achieving an improved fit between the projected model and the image features, which was the main motivation for implementing the model-to-image matching in this study. The first method, which uses RANSAC and the Gauss-Markov model, performs well for well geo-referenced images. The camera positions and rotations are have accuracy of $1-3$ [m] and $0.1-0.3°$, respectively (Tab. 8.2). Results on the second method, which performs matching using an accumulator and the Gauss-Helmert model, are good, even if the exterior orientation of the camera has accuracy of 7 [m] and $0.7°$.

Fig. 8.4b illustrates an improvement in the position of the projected model compared to the direct geo-referencing. The projected building structures match the TIR image very well, so the thermal building textures can be extracted precisely. Nevertheless, the accuracy of the estimated exterior orientation parameters is not very high because of the low resolution of the images and the low accuracy of the extracted edges. The presented method, in combination with this acquisition configuration, cannot be used for precise determination of the camera positions; however, this was not the goal of this work.

In TIR images of urban scenes it is difficult to extract building edges on the ground, since the pavement often looks similar to walls. Therefore, in some frames, a very good fit between the 3D building model and the roof structure was achieved, while some remaining displacements occurred in the walls.

Elimination of outliers: By applying the RANSAC or accumulator algorithm, good results can be achieved, not only for very well geo-referenced data, but also for mid-quality geo-referencing. However, further improvements in the search for candidate correspondences are needed in case of low-quality input, which should be subject of future studies. The accumulator is sensitive to inaccuracies in the angles because inaccuracies of a 3D to 2D projection are approximated by a 2D motion (translation and rotation) of the projected 3D model. This assumption holds only for the very well known rotation parameters of the camera. Small inaccuracies in the angles of the exterior orientation cause a shift in the projected model; however, an increase in angle errors deforms the projected 3D model, which does not allow the accumulator approach to work efficiently. The proposed RANSAC algorithm does not depend on the angle errors in the

initial camera pose, but large errors in the initial rotation increase the search area and, in many cases, can lead to lines being falsely assigned.

Tests showed that matching using the accumulator and the Gauss-Helmert model is more reliable and delivers better results, even when initial exterior orientation of the camera is less accurate, than using RANSAC and the Gauss-Markov model. On one hand this is due to the slightly better performance of the accumulator algorithm in comparison to RANSAC (see Tab. 8.1). On the other hand it is also related to the fact that the instabilities detected in the Gauss-Markov model when using vertical lines are avoided in the Gauss-Helmert model.

The accumulator performs better in the presented application because it works systematically, taking all correspondences into account. RANSAC only investigates a part of the population and makes the decision based on statistical evaluation, which can cause errors. The accumulator is also faster because the matching problem is reduced to 2D.

Optimal pose estimation: Taking the uncertainties of image lines and of the building model into account allows us to apply a statistical analysis based on uncertainties, such as statistical tests and robust estimation with outlier detection. A better fit between the building model and the image structures is also achieved.

Each optimization results in new exterior orientation parameters. However, these parameters must not be understood as the true position of the camera, only as the parameters needed for a better fit between the features and the edges of the 3D building model.

Both presented adjustment models, one formulated using the Gauss-Markov and the other formulated using Gauss-Helmert model, provide corrections for the model edges. However, these corrections cannot be directly applied to improve the geometry of the model because no constraints for planes are implemented in the presented adjustment. Therefore, some coplanar 3D building lines, which originally belonged to one plane, can be non-coplanar after the adjustment.

Exterior orientation parameters estimated based on the matched lines are used to find an improved fit between the model and the image structures, which was the main motivation of this study. These parameters, however, do not necessarily capture the correct position and orientation of the camera in the real world. In Fig 8.9, the estimated parameters do not build a smooth trajectory as the input data that was obtained with the GPS/INS system and smoothed by a Kalman filter. Some of the "jumps" of the camera are the result of invalid co-registrations, and perhaps are also caused by a connection between the camera and the navigation device if it was not fully rigid and suffered from the vibrations of the helicopter. The correctness of the estimated parameters could be assessed by tracking the helicopter using terrestrial measurement in an independent coordinate system, e.g. by tracking a target on the helicopter with a tachymeter.

Conditioning: The experiments showed, that conditioning has a strong influence on the matching result. In Fig 8.9, the estimated parameters Z_0 and ϕ differ significantly from the initial value. An error of about 25 [m] and 1.5° in the initial orientation parameters, which would be the result of adjustment without conditioning, is not realistic. The result of the adjustment with conditioning oscillates around the initial values and is therefore more likely to be correct.

It was also shown that conditioning is crucial for achieving a better fit between the image and the building features, which was the main motivation of this work. Fig. 9.2 shows an exemplary frame, where the effect is highly visible, particularly in the boundary areas. In this figure, the conditioning procedure with various $f_{\min}$ factors is presented. It can be seen that estimation without conditioning leads to a large mismatch in the boundary area (Fig. 9.2a). This mismatch

is reduced by conditioning; however, the value of $f_{\min}$ also changes the result. The best fit was achieved with $0.1 \leq f_{\min} \leq 1$.
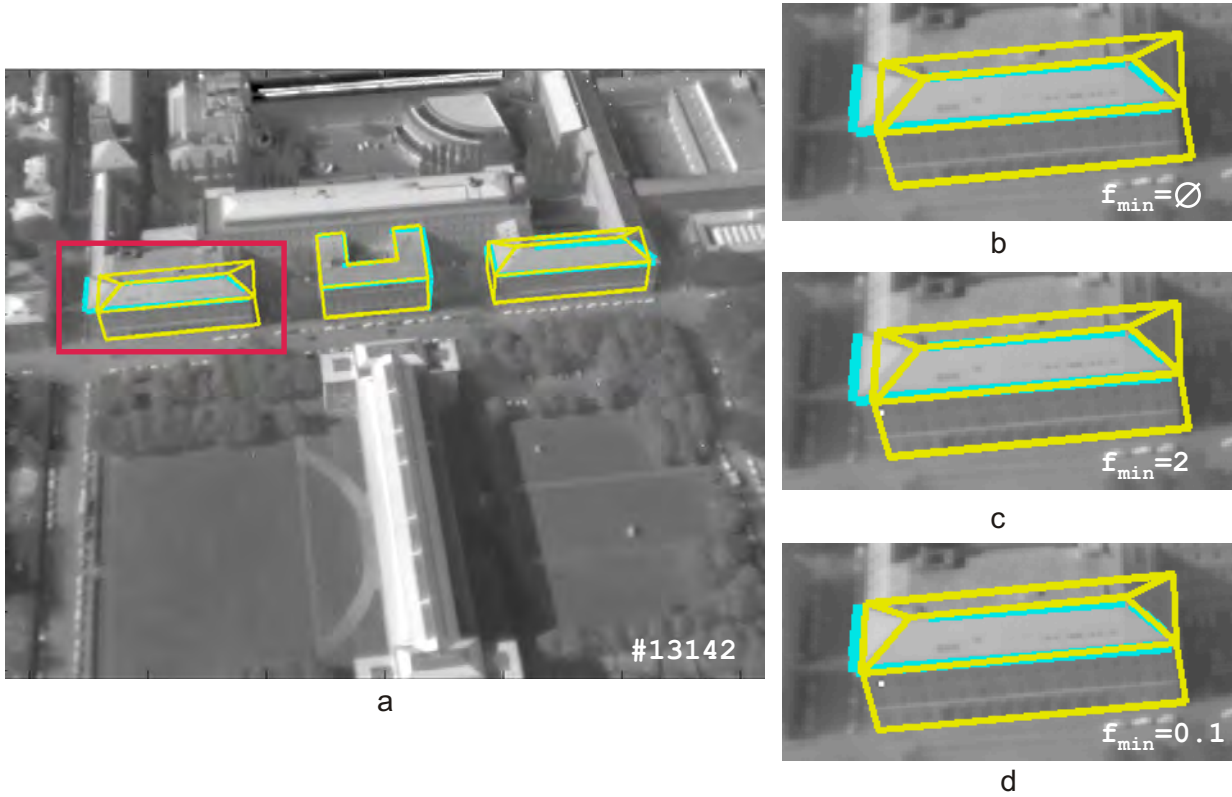

Figure 9.2: Matching results with various $f_{\min}$ factors: a) scene overview; b) no conditioning; c) conditioning with $f_{\min} = 2$; d) conditioning with $f_{\min} = 0.1$. Color coding: yellow - projected 3D building model after estimation; cyan - extracted image line segments, which where used for the estimation.

Robust estimation: Inaccurately extracted line segments or inaccurate model edges can be eliminated using the outlier detector within the adjustment procedure (as described in Section 4.3). In Fig. 9.3, some examples of eliminated line correspondences are presented. In most cases, the reason for the rejection is inaccurate line segment extraction.

In Fig. 9.3a, an image section of building B3 is presented. The roof ridge of this building was assigned to two line segments in the image, but one of them is a false assignment. The outlier detection algorithm removed this correspondence. An eave and a vertical edge (on the left side of building B3 in Fig. 9.3a) were assigned to the same line segment in the image. This line segment, however, was detected inaccurately in the image and could not accurately match any of the assigned models edges, which is why this correspondence was rejected.

In Fig. 9.3b, building B3 is seen in an another frame. Here, the ridge was poorly detected in the image and the correspondence was rejected. Similarly, in Fig. 9.3c, which shows the building B2, one edge was also poorly detected and rejected due to this reason. Fig. 9.3d shows building B2 again; however, here, more correspondences have been rejected, because in this frame the extraction of some segments was inaccurate.

In the Gauss-Markov model with pseudo-observations, errors in the model lines are difficult to recognize using the coefficient matrix of corrections to observations (Q_{vv}). This is due to the fact that their redundancy components are low, and therefore the influence of the errors on
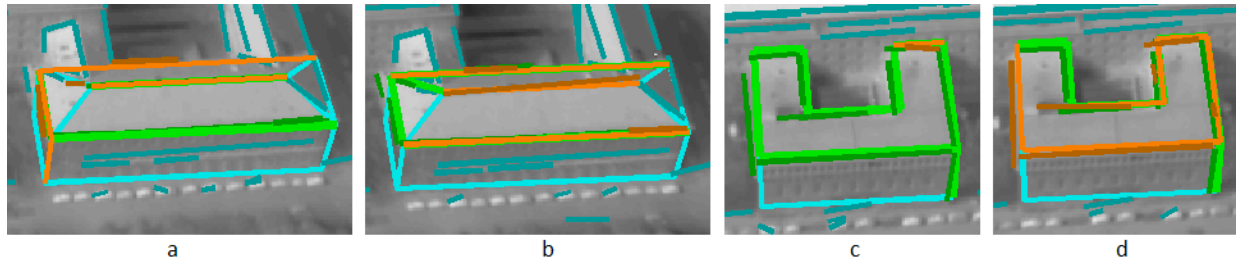
Figure 9.3: Examples for line correspondences identified as outliers in the robust adjustment: a) building B3, frame #13204; b) building B3, frame #13206; c) building B2, frame #13204; d) building B2, frame #13205. Color coding: light cyan - projected model edges; dark cyan - extracted line segments; green - accepted correspondences (dark - image line segments, light - model edges); orange - correspondences identified as outliers (dark - image line segments, light - model edges).

the correction for each parameter is very small. In the Gauss-Helmert model, the redundancy components for the 3D points are significantly larger, and they can be used to detect potential outliers in the 3D building model.

Tracking: By tracking the line segments assigned to the 3D model from frame to frame, the search area is restricted, and the time needed for calculation is reduced. Up to now, line tracking experiments have been conducted with pre-defined key-frames. In the future, more attention should be paid to dynamically selected key-frames and to the criteria for coregistration reliability in a single frame. The following frame in the sequence can be set to a key-frame (in case of low reliability) or to standard frame (in case of high reliability).

The tests showed that tracking is sufficient for finding the line correspondences needed for camera pose estimation. Two cases were considered: in the first case, the estimation was carried out in *key-frames* only (Tab. 8.3), and in the second case the estimation was carried out in every frame (Tab. 8.4). In the first case, the quality measure for tracking was significantly higher when the distance between the frames was increased. This dependency was not noticed in the second case, where the adjustment was carried out in every frame.

Due to a high computation time in *key-frames*, it is plausible to search for correspondences in the first frame, and then to continue with tracking only. The *key-frames* could still be used to control the quality of tracking in this scenario, which would reduce the computational time to about 0.1 [sec] per frame. This, however, is not sufficient for real time tracking. Assuming a frame rate of 25 [frames/sec], the computation time must not exceed 0.04 [sec]; therefore, a real time solution would require optimizing the software.

Tracking also produced a kind of a database which was created by linking image lines to each other from frame to frame. Because the lines were also assigned to the 3D building models, there is also a link between the lines observed from multiple directions. Such information can be used to improve and refine the geometry of the existing 3D building model.

9.2 Discussion on Texturing

Best texture selection: The algorithm for best texture selection chose good quality textures for most model faces. As expected, the textures have been selected from all four directions (Fig. 8.14), depending on the quality calculated for each separate face. It was not possible to find high quality textures for a few surfaces (Fig. 9.4), namely on small faces, such as the side surfaces of chimneys and ventilation systems (Fig. 9.4b, c, and f). The disadvantageous orientation of the

face and occlusion, of these particular surfaces could explain low texture quality. However it is also possible, that the low quality is a result of a calculation inaccuracy for small faces, which is related to the inaccurate approximation of occupied area after the projection. This is calculated based on pixel count occupied by a projected model surface. This procedure can be inaccurate for long, narrow faces.
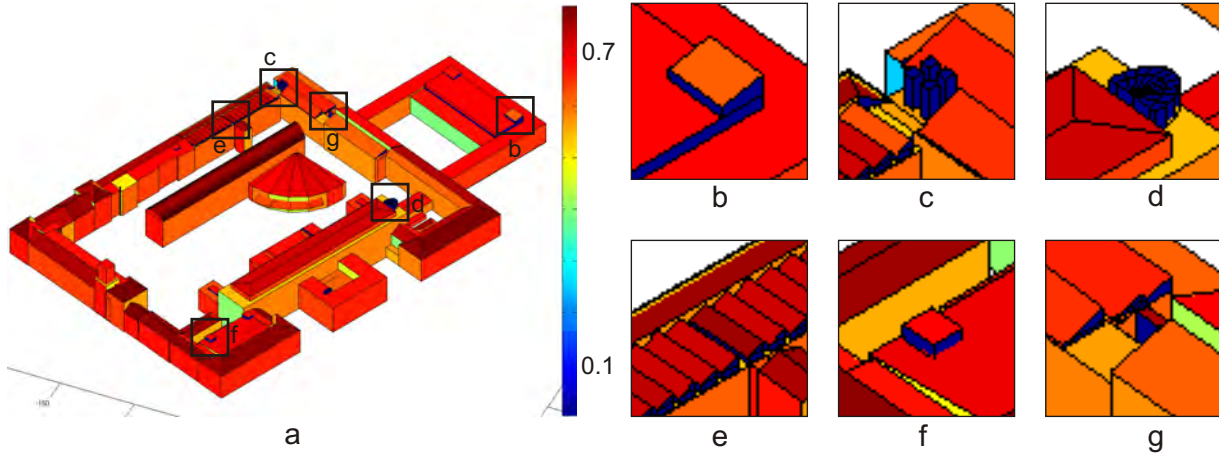

Figure 9.4: Surfaces with low quality textures: a) overview; b-g) examples of surfaces with low quality textures (blue)

Local Matching: In most cases, local matching makes it possible to overcome the errors in the model and the extracted textures, which are depicting the whole façade. Hence, no part of the façade is missing and no additional objects are included in the texture. This rule does not apply for façades modeled as rectangles, which have other shapes in reality. An example of such a façacde is presented in Fig. 9.5.
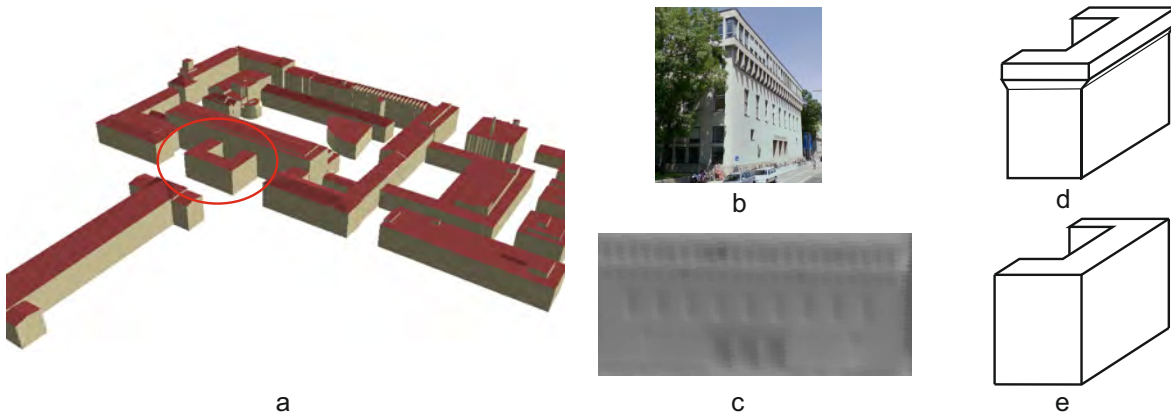
Figure 9.5: Example of a building with incorrectly modeled façade geometry: a) localization of this building in the data set; b) RGB image of this building, c) TIR texture for the frontal façade; d) schematic model of this building; e) model used for the texture extraction

This façade belongs to a building that is placed in the front section of the test area (Fig. 9.5a or building B2 in Fig. 7.1). A VIS image of this building is presented in Fig. 9.5b. It can be observed, that lower level of the building is slimmer than the top story, making it necessary to model it as presented in Fig. 9.5d. However, the building models used for the experiments were created using aerial images. Pre-defined shapes were fitted to the roof shape seen in stereo mode and extruded

these shapes to the ground. This result of extracting a slightly incorrect texture before local matching is shown in Fig. 9.5c. In this case, the left side of the texture was adjusted to the lower edge of the façade, and the right side of the texture was adjusted to the upper edge. On the left side, the upper part of the façade is depicted completely in texture, while in the lower part of the texture, an object not belonging to the façacde occupies a small stripe of pixels. In contrast, on the left side, where the edge was adjusted to the lower part, there is no additional stripe of pixels belonging to any other objects, but there is a small part of the building missing in the upper part of the texture. An error free extraction of textures is not possible in such cases without correcting the 3D building model. Automatic model correction, however, is a complex task, especially for such buildings as presented in in Fig. 9.5d. Particularly, the automatic detection of such buildings can be difficult. In such buildings, certain sections of the building can be relatively low, causing short edges, which are detected with low accuracy. Because of the lack of accuracy, these edges cannot be used for automatic improvement of the model geometry. Automatic detection and correction of the building models, where the roof overlap is missing (see Fig. 10.1), is generally easier to implement.

Texture extraction: Another common and challenging source of errors that can occur during texture extraction are un-modeled (non-predictable) occlusions. These occlusions occur when some objects, such as vegetation, cars, city furniture, etc., are not included in the model or are not modeled correctly. Some examples of non-predictable occlusions in the test area are presented in Fig. 9.6. Fig. 9.6a shows the location of two exemplary textures in the model (red frame); Fig. 9.6b and Fig. 9.6c show those textures. The texture presented in Fig. 9.6b is occluded by two chimneys, which are not correctly modeled. The texture presented in Fig. 9.6c is occluded by trees, which are also not included in the 3D building model. In both cases, an adaption of existing algorithms to detect non-predictable occlusions [Böhm, 2004; Abdelhafiz & Niemeier, 2009; Abdelhafiz, 2009; Bénitez et al., 2010] could be implemented.
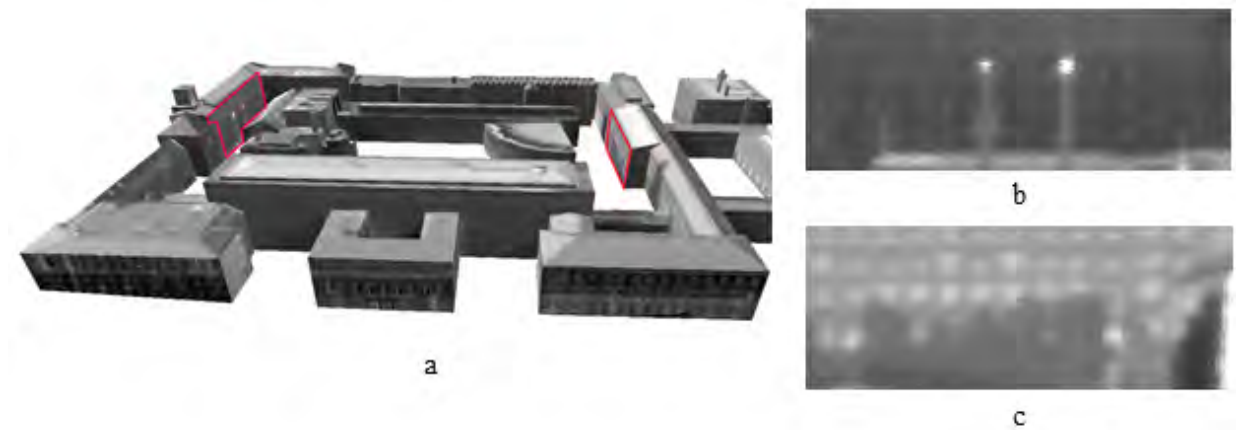

Figure 9.6: Examples of textures with external occlusions: a) test area with textures with highlighted examples of non-predictable occlusions; b) texture occluded by chimneys; c) texture occluded by trees.

9.3 Discussion on Windows Detection

The initial results on window detection in airborne thermal textures showed that the quality of those textures, particularly in terms of contrast and resolution, as well as smoothed edges, make detection difficult. Using local dynamic threshold, it was possible to find a sufficient number of candidates (Tab. 8.8), but the algorithm did not succeed in confirming them with the corner

detector (masked correlation). Only the detection in Façade 4 was successful because this façade is directed to the south, and the measurement was carried out in the early afternoon. The contrast was increased because the sun reflected off the window panes.

In contrast to the results of the airborne textures, detecting windows in the terrestrial textures was more successful. Here, the local dynamic threshold made it possible to extract appropriate candidates for windows. This segmentation technique helped to bridge the different appearances caused by various objects reflected in window panes. Masked correlation and geometric constraints made it possible to refine the position of each window. Using the mask, which switches off irrelevant parts of the window, and searching for intensity changes in the image around the window corners delivers promising results; however, the mask applied in this research did not match all possible window shapes. Occlusions by trees, traffic signs, and lights, as well as by other buildings, also made window detection difficult. Thus, more research on occlusion free texture extraction is needed.

The achieved results of window detection in terrestrial TIR textures were satisfying: about 70% completeness and correctness. However, the results on window detection presented in this thesis should be seen as pre-processing step, which can still be improved using detection methods based on stochastic processes. The low completeness of the results (26% for airborne data and 66% for the terrestrial data) means that several windows remained undetected. This would be disadvantageous for visual perception because irregular structures can be easily noticed by human observer (8.18). However, the goal of window detection in this thesis was to identify and exclude areas that are useless for thermal inspections. Detected windows can be used to indicate heat leakage from radiators, which are often placed below the window. With regard to this goal, even a few detected windows can be helpful for further analyzing of the thermal textures.

The size of the detected windows was overestimated in most cases (Tab. 8.10). This can be due to the fact, that some windows are framed by another material that can be recognized as window frame by the proposed algorithm.

10 Conclusions and Outlook

In this chapter, the thesis is summarized, the contribution of the presented work is emphasized and potential topics for future work are outlined.

10.1 Summary

This dissertation has investigated the potential of utilizing thermal infrared image sequences taken from a flying platform in urban areas for building texturing and its application for the thermal inspection of buildings. The main objective has been to develop a methodology that makes it possible to extract the best quality textures by emphasizing the geometric quality of the textures. Therefore, the most important part of the thesis has elaborated on the co-registration of 3D building models and TIR images. The presented methodology is suitable for single images, but it can also be used to take advantage of image sequences. For this purpose, a tracking strategy has been developed. Since a good fit between the textures and the model does not depend only on the co-registration, but also on the geometric accuracy of the model, typical problems and errors in 3D building models were discussed. A solution for overcoming model inaccuracies using local matching for each edge separately has been suggested; however, as alternative option, improving the model's geometry has been considered. Some attention has also been paid to best texture selection where the main indicators for the quality are resolution and distortion due to an oblique view. Quality measures assessing the matching quality have also been introduced. In addition, a first attempt to detect objects in the thermal textures has been presented and discussed.

The tests showed good results on co-registration, particularly in cases where tracking between the neighboring frames had been applied. Local matching also improved the fit between the image features and the model edges which enabled better fitting textures to be extracted. However, these extracted textures have a low resolution and contrast, which, unfortunately, makes it difficult to use them for object detection which would be an important contribution to automatic thermal building inspection.

10.2 Contributions of the Work

Thermography: This work contributes to thermography by investigating the usefulness of thermal imagery taken from a flying platform for the thermal inspection of buildings.

Photogrammetry: This work contributes an innovative formulation of the model-to-image problem using linear features and projective geometry. This innovation adapts the estimation method proposed by Meidow et al. [2009b] in terms of the definition of an appropriate functional and stochastic model. In addition, the optimal pose estimation proposed by Meidow et al. [2009b] was extended using an outlier detection algorithm.

3D Building Modeling: This work contributes to building modeling in terms of the enrichment of 3D building models with thermal textures. Moreover, a concept for storing some information, such as permanent occlusion in the form of binary textures was also presented. Finally, possible ways to store geometry of neighboring building and the most advantageous solution for texture mapping was selected and implemented in *CityGML* code were discussed.

At the beginning of the thesis, three research questions were asked.

Will co-registration results significantly improve if uncertainties of 3D models and of image features are considered in the estimation?
The results of this study did not show significant improvement in the fit between the projected model and the image when uncertainties were considered in the estimation. The fit, however, was improved by conditioning the coordinates in the method using homogeneous coordinates and the Gauss-Helmert model. Additionally, considering uncertainties in the estimation is advantageous because the stochastic model can be formulated, making it possible to implement the outlier detector within the estimation and remove faulty observations. The co-registration results strongly depend on the correctness of the correspondences. Taking uncertainties into account makes it possible to remove incorrect correspondences in an effective, universal, and mathematically correct way. Considering uncertainties in the estimation, therefore, indirectly improves the co-registration results.

To what extent does line tracking support model-to-image matching?
Tracking lines from frame to frame facilitates faster model-to-image assignments than running the whole procedure of model-to-image matching in each frame. Tests showed that even a simple tracking method using cross correlation is reliable enough to estimate the exterior orientation parameters of the camera based only on the tracked lines. The full model-to-image matching has to be carried out in the first frame and then in the key-frames. The experiments did not show a dependency between the key-frame interval and the accuracy of the matching. The interval, therefore, can be set relatively high (20 frames or more). Using the key-frames for initialization, control, and quality assessment could also be considered. The camera pose estimation could only be conducted by line tracking; however, this can only be achieved in cases when the overlap between the frames is very large.

What is the potential for the textures extracted from airborne thermal images to be used for window detection?
Window detection in airborne thermal textures is a challenging task, as seen in the first results. Two problems arose in this study. First, it was difficult to find correct candidates for windows using the segmentation with a local dynamic threshold. Second, extracting window corners and window edges in order to find the exact shape of the window was difficult because of the smoothed edges, as well as the low contrasts and low resolutions of the airborne thermal textures. The correctness of 63% is a moderate success for the first results, but the completeness of 26% falls below expectations. Moreover, it is important to point out that these values differ significantly, depending on the façade. The detection showed much better results for façades that were exposed to the sun than façades that were facing another direction. The higher degree of completeness and correctness for façades directed towards the sun is related the fact that the window panes reflect light. Although sun reflections are undesirable for thermal inspections, they helped to better detect the windows in this experiment. Tests with terrestrial data showed that correctness and completeness could be improved by using a higher resolution of the textures; however, this method

also contains unsolved problems, limiting the completeness of the detection to 50%. Overcoming these difficulties should be subject of future work.

10.3 Future Work

The findings of this thesis have yielded a number of potential topics for further research:

- *Visibility check for lines:* In individual cases, the current implementation the visibility check algorithm for lines leads to the incorrect identifications of invisible lines related to numerical errors. In the future, more research on how to avoid this problem is needed. Moreover, an algorithm to remove densely projected lines should be developed in order to avoid wrong line correspondences and to reduce unnecessary computational effort.

- *Tests with other platforms:* Using a helicopter to acquire data allows for more payload, which means larger cameras and the combination of many sensors, but usage of this platform is expensive and requires complicated procedures before the flight departs. In the last few years, unmanned aerial vehicles (UAVs) have become more and more popular. There are already good quality TIR cameras available that can be mounted on a UAV and used to thermally inspect a building. UAVs are flexible in terms of acquisition trajectory and they can be flown closer the building so thatthe whole façade can be taken in one picture. This makes texture extraction easier than in case of terrestrial imagery.

- *Tests in other scenes:* The developed methodology is not restricted to outdoor scenes. It should also be tested in indoor environments. Indoor thermal textures would allow for advanced investigations of buildings, making it possible to understand the complex dependencies of heat circulation in a building. Also, combining indoor and outdoor thermal textures with a 3D GIS including additional information of the buildings such as heating supply data [Kaden & Kolbe, 2013] could be investigated.

- *Tests with other sensors:* The presented methodology for co-registration could also be adapted for visible data. This would make it possible to test this methodology in further applications, such as indoor navigation or autonomous flying. Also, combining multiple data sets with 3D models can enhance scope of analysis and object detection.

- *Improvements of co-registration and tracking:* The presented method can only deal with geo-referenced data; therefore, in the future, a strategy for analyzing data without the initial position information should be developed. For this purpose, calculating vanishing points can be utilized in order to recover the orientation angles. Also, relational matching can be tested as a method to help find approximated exterior orientation parameters. Neighboring frames should be employed in the adjustment to improve the relative orientation of the sequence. This should stabilize the model-to-image matching by reducing the movements of the camera.

- *Rolling shutter:* In the future, more attention should be paid to the rolling shutter effect. This phenomenon should also be taken into consideration in the adjustment. Typically, the TIR cameras read out the measured radiation row-wise. If the camera is moving, each row is read at a different point in time, and therefore at a different position, which should be accounted for in the geometrical model used for estimation.

- *Improvements in model geometry:* Line tracking through the image sequence delivers correspondences between the line segments in different frames. The links between the image lines in neighboring frames should be used to improve the geometry of the 3D building models. This can be achieved by formulating conditions for 3D planes based on the fact that two crossing 3D planes $\mathbf{A}_1$ and $\mathbf{A}_2$ produce a 3D line $\mathbf{L}$ such that $\mathbf{L} = \mathbf{A}_1 \cap \mathbf{A}_2$, where $\cap$ is the meet operator. The projection can then be written as $\mathsf{Q}(\mathbf{A}_1 \cap \mathbf{A}_2) = \mathbf{l}$, where Q is the projection matrix for lines. This condition, together with parallelism and orthogonality conditions for planes, can be used to estimate the plane parameters in the 3D model, and improve the 3D model as shown in Fig. 10.1. However, defining the building type and all the conditions automatically is very challenging, especially for complex scenes, such as the presented test area because buildings that do not share the same shape are merged and there is a high grade of occlusion.
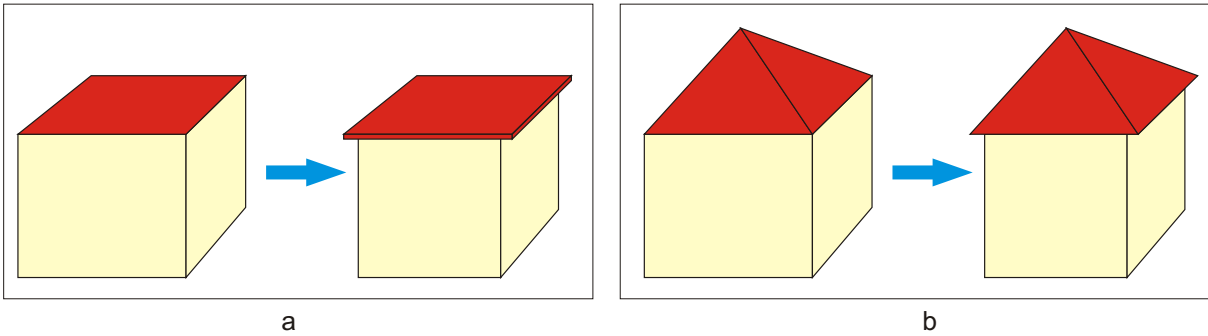
Figure 10.1: Improvement of the 3D geometry of the building models: a) flat roof; b) hip roof.

- *Un-modeled occlusions:* Further challenges include occlusions caused by vegetation or un-modeled buildings; therefore, in the future, the exterior occlusions should be detected and handled beside the self occlusion caused by the 3D building model. The point cloud created in the *Structure-from-motion* algorithm or acquired by a laser scanner can be utilized for this task.

- *Improvements of window detection:* In further studies, the presented method for window detection should be integrated into a stochastic process for determining the probabilities of window positions. The presented geometric constraints could then be described by a likelihood, which would also allow exceptions from a typical case. By applying this solution, rows with different window sizes could also be modeled.

There were already attempts to improve the results presented in this thesis for the terrestrial textures using a production system [Michaelsen et al., 2012]; however, due to the geometric distortions of the textures, the applied methodology based on a GESTALT-system [Michaelsen, 2004] failed to detect most of the windows (Fig. 10.2). In this study, two production systems, "window first" (Fig. 10.2b) and "row first" (Fig. 10.2c), were implemented. Both systems, however, delivered similar results.

The motivation for window to be able to identify regions that are not useful for thermal inspection because of light reflecting off window panes. In such a case, the precise size and shape of the detected windows are not as important as the fact that false hot spots on the window pane are excluded from further processing. Also, the correctness of the results is more important than the completeness. If a window was skipped by the detector due to the similarity to the background, it does not influence further processing. It is important, however, that the other hot spots, which are of particular importance for the thermal inspections, are not recognized as windows. Especially challenging is separating the hot spots reflected off the window pane and
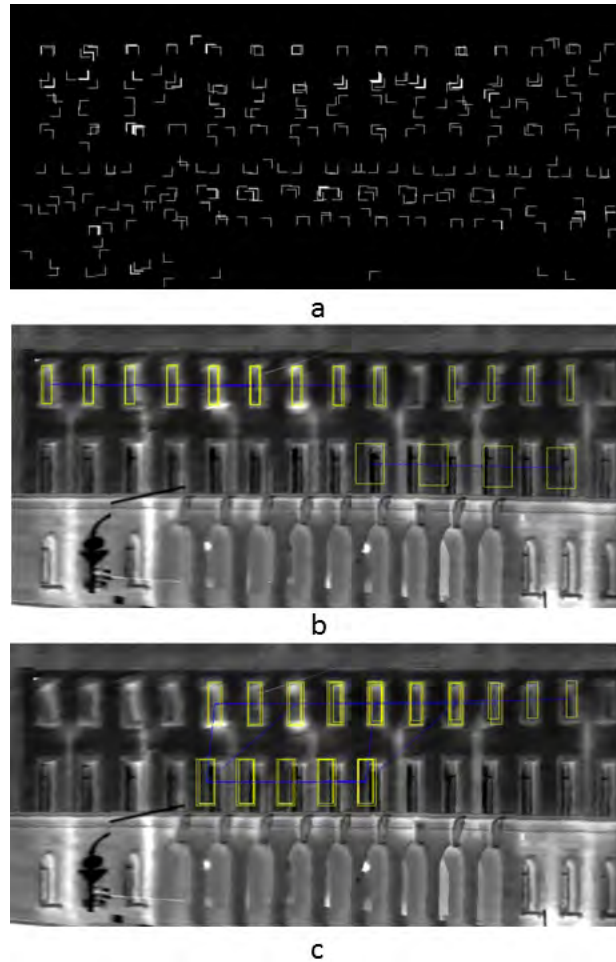
Figure 10.2: Detection of windows using a production system: a) extracted primitives (corners); b) result with "windows first" productions; c) result with "rows first" productions

the hot spots below the window. To prevent classifying these regions as part of the windows, the window model could be extended with a heating in the bottom. The façade could then be classified simultaneously in a window class and a heating-hot-spot class by taking the contextual information of the scene into account.

Another possibility for improving the results of window detection is to use laser point clouds to determine candidates for windows [Tuttas & Stilla, 2011]. Moreover, the window model should be extended so that more shapes are allowed. For example, arcs in the upper part of the window were not considered in this research. This shape, however, is a common structure, particularly in old buildings, which are often objects of thermal inspections.

- *Super-resolution:* Co-registration and model projection can offer accurate information about the 3D coordinates of each pixel. Since every façade can be seen multiple times, this information can be used to calculate super-resolution textures. In the future, their usability for object detection should be investigated.

References

Abdelhafiz A (2009) *Integrating digital photogrammetry and terrestrial laser scanning.* PhD thesis, Institute for Geodesy and Photogrammetry, Technical University Braunschweig, Germany.

Abdelhafiz A, Niemeier W (2009) Integration of digital images and laser scanning point clouds - texture mapping with 3DImage. *Journal of DVW*, 60: 113–125. 91. DVW-Seminar Terrestrial laser scanning TLS2009 Fulda, Germany, November 19-2.

Allen P, Stamos I, Gueorguiev A, Gold E, Blaer P (2001) AVENUE: Automated site modeling in urban environments. In: *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*: 357–364.

Avbelj J, Iwaszczuk D, Stilla U (2010) Matching of 3D wire-frame building models with image features from infrared video sequences taken by helicopters. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-B: 149–154. Proceedings of PCV 2010 - Photogrammetric Computer Vision and Image Analysis.

Baden S, Fairey P, Waide P, Laustsen J (2006) Hurdling financial barriers to lower energy buildings: Experiences from the USA and Europe on financial incentives and monetizing building energy savings in private investment decisions. In: *Proceedings of ACEEE Summer Study on Energy Efficiency in Buildings, American Council for an Energy Efficient Economy*

Balaras C, Argiriou A (2002) Infrared thermography for building diagnostics. *Energy and Buildings*, 34: 171–183.

Becker S (2009) Generation and application of rules for quality dependent facade reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64 (6): 640–653.

Becker S, Haala N (2009) Grammar supported façade reconstruction from mobile LiDAR mapping. In: Stilla U, Rottensteiner F, Paparoditis N (eds) *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (3/W4). Proceedings of CMRT09. Paris, France.

Beder C (2004) Fast statistically geometric reasoning about uncertain line segments in 2D- and 3D-space. In: C.E.Rasmussen, H.H.Bülthoff, B.Schölkopf, M.A.Giese (eds) *Proceedings of the DAGM Symposium* (3175): 375–382.

Beder C (2007) *Grouping uncertain oriented projective geometric entities with application to automatic building reconstruction.* PhD thesis, Photogrammetry Department, Institute for Geodesy and Geoinformation, Bonn University, Germany.

Bénitez S, Baillard C (2009) Automated selection of terrestrial images from sequences for the texture mapping of 3D city models. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-3/W4: 97–102. Proceedings of CMRT09. Paris, France.

Bénitez S, Denis E, Baillard C (2010) Automatic production of occlusion-free rectified façade textures using vehicle-based imagery. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-3A: 275–280. Proceedings of PCV 2010 - Photogrammetric Computer Vision and Image Analysis.

Berg A, Ahlberg J (2014) Classification of leakage detections acquired by airborne thermography of district heating networks. In: *Proceedings of the 8th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS), Stockholm*

Blackledge J (1997) *Image processing: Mathematical methods and applications.* Oxford University Press. ISBN: 978-0198511977.

Bochko V, Petrou M (2007) Recognition of structural parts of buildings using support vector machines. In: *Proceedings of Pattern Recognition and Information Processing, PRIP2007*

Böhm J (2004) Multi-image fusion for occlusion-free façade texturing. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35-B5: 867–872. Proceedings of XX ISPRS Congress, Istanbul, Turkey.

Bornik A, Karner K, Bauer J, Leberl F (2001) High quality texture reconstruction from multiple views. *Journal of Visualisation and Computer Animation.*

Borrmann D, Elseberg J, Nüchter A (2012) Thermal 3D mapping of building façades. In: *Proceedings of the 8th Conference on Intelligent Autonomous Systems* Jeju Island, Korea.

Boyd G (2013) Zeroing in on energy savings with thermal imaging. *Photogrammetric Engineering & Remote Sensing*, 79 (4): 313–316.

Bronzino JD, Diakides NA (2008) *Medical infrared imaging.* CRC Press. ISBN: 9780849390272.

Cabrelles M, Galcera S, Navarro S, Lerma JL, Akasheh T, N. H (2009) Integrationof 3D laser scanning, photogrammetry and thermography to record architectural monuments. In: *Proceedings of the 22nd CIPA Symposium, Kyoto, Japan,*

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

Cham T, Ciptadi A, Tan W, Pham M, Chia L (2010) Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, USA, 2010*

Chandler D (2011) The big picture on energy loss. MIT news `http://web.mit.edu/newsoffice/2011/ir-scanning-energy-0316.html`. Access on July 30th, 2012.

Cheng P, Anderson M, He S, Zakhor A (2013) Texture mapping 3D planar models of indoor environments with noisy camera poses. In: *Proceedings of SPIE Electronic Imaging Conference, Multimedia Content Access, Burlingame, California*

Cothren JD (2005) *Reliability in constrained Gauss-Markov models : an analytical and differential approach with applications in photogrammetry.* Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Technical report.

Debevec P, Taylor C, Malik J (1996) Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*: 11–20.

Dick A, Torr P, Cipolla R, Ribarsky W (2004) Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60 (2): 111–134.

Ding M, Zakhor A (2008) Automatic registration of aerial imagery with untextured 3D LiDAR models. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*

Drauschke M (2011) *Ein hierarchischer Ansatz zur Interpretation von Gebäudeaufnahmen.* PhD thesis, Institute of Photogrammetry, University of Bonn.

ESRI (2015) CityEngine. Online: `http://www.esri.com/software/cityengine`. Access on September 11th, 2015.

Eugster H (2011) *Echtzeit-Georegistrierung von Videodaten mit Hilfe von Navigationssensoren geringer Qualität und digitalen 3D-Landschaftsmodellen.* PhD thesis, Mathematische-Naturwissenschaftliche Fakultät II der Humboldt-Universität zu Berlin.

Eugster H, Nebiker S (2007) Geo-registration of video sequences captured from mini UAVs approaches and accuracy assessment. In: *Proceetings of 5th International Symposium on Mobile Mapping Technology - Padua, Italy*

Eugster H, Nebiker S (2008) UAV-based augmented monitoring - real-time georeferencing and integration of video imagery with virtual globes. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 37-B1: 1229–1235. Proceedings of XXI ISPRS Congress, Beijing, China.

Eugster H, Nebiker S (2009) Real-time georegistration of video streams from mini or micro UAS using digital 3D city models. In: *Proceedings of 6th International Symposium on Mobile Mapping Technology*

Faugeras O, Luong QT, Maybank S (1992) Camera self-calibration: Theory and experiments. In: Sandini G (ed) *Computer Vision - ECCV'92*, volume 588 of *Lecture Notes in Computer Science* (pp. 321–334). Springer Berlin Heidelberg.

Feldmeier F, Rossa M (2009) *Die Bauthermographie. Gebäude in anderem Licht.* Rosenheim, Ifz.

Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6): 381–395.

Fondazione Bruno Kessler (2014) Non-active project at 3DOM - 3D Optical Metrology Unit: IR3D (2008-2009). Online: `http://3dom.fbk.eu/en/node/89`. Access on June 5th, 2014.

Förstner W (1979) Das Programm TRINA zur Ausgleichung und Gütebeurteilung geodätischer Lagenetze. *ZfV - Zeitschrift für Vermessungswesen*, (2): 61–72.

Förstner W (2004) Uncertainty and projective geometry. In: Bayro-Corrochano E (ed) *Handbook of computational geometry for pattern recognition, computer vision, neurocomputing and robotics* (pp. 493–535). Springer.

Förstner W (2010a) Minimal representations for uncertainty and estimation in projective spaces. In: *Proceedings of Asian Conference on Computer Vision*: 619–633, Part II. Queenstown, New Zealand.

Förstner W (2010b) Optimal vanishing point detection and rotation estimation of single images of a legolandscene. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-3A. Proceedings of ISPRS Commission III Mid-Term Symposium, Paris, France, September 1st-3rd.

Förstner W (2012) Minimal representations for testing and estimation in projective spaces. *PFG Photogrammetrie Fernerkundung Geoinformation*, 3: 209–220.

Förstner W, Korč F (2009) *eTRIMS image database for interpreting images of man-made scenes.* Department of Photogrammetry, University of Bonn, Technical report. (TR-IGG-P-2009-01).

Förstner W, Neumann B, Šára R, Petrou M, Hotz L (2009) eTRIMS - e-training for interpreting images of man-made scenes. Online: `http://www.ipb.uni-bonn.de/projects/etrims/`. Access on July 10th, 2013.

Förstner W, Wrobel B (2016) *Photogrammetric computer vision. Geometry, Orientation and Reconstruction.* Springer. To appear in 2016. ISBN 978-3-319-11550-4.

Fouad NA, Richter T (2012) *Leitfaden Thermografie im Bauwesen. Theorie, Anwendungsgebiete, praktische Umsetzung.* Fraunhofer IRB Verlag. ISBN: 978-3816770718.

Fraser CS (1997) Digital camera self-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 52 (4): 149 – 159.

Früh C, Sammon R, Zakhor A (2004) Automated texture mapping of 3D city models with oblique aerial imagery. In: *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04)*

Gaussorgues G (1994) *Infrared Thermography*, volume 5 of *Microwave Technology Serien.* Chapman and Hall. ISBN: 978-94-010-4306-9.

Gonzalez R, Woods R (2008) *Digital image processing.* Prentice Hall, 3rd edition. ISBN: 978-0131687288.

Götz C (2010) Bestimmung der Layover-Bereiche bei der SAR-Abbildung von Gebäudemodellen. Master's thesis, Fachgebiet Photogrammatry and Remote Sensing, Technische Universität München.

Grabner H, Šochman J, Bischof H, Matas J (2008) Training sequential on-line boosting classifier for visual tracking. In: Borgefors G, Flynn P (eds) *Proceedings of the 19th International Conference on Pattern Recognition*

Green PJ (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82: 711–732.

Grenzdörffer G, Guretzki M, Friedlander I (2008) Photogrammetric image acquisition and image analysis of oblique imagery. *The Photogrammetric Record*, 23 (124): 372–386.

Grewal M, Weill L, Andrews A (2007) *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons. ISBN: 978-0-470-04190-1.

Gröger G, Kolbe T, Nagel C, Häfele KH (2012) OGC City Geography Markup Language (CityGML) encoding standard.

Gröger G, Plümer L (2012) CityGML - Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71 (0): 12 – 33.

Groneman A (2004) TOPOSCOPY combines photogrammetric modelling with automatic texture mapping. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 34-5/W12: 168–172. Proceedings of the ISPRS workshop on Vision techniques applied to the Rehabilitation of City Centres held in Lisbon, Portugal.

Haala N, Kada M (2010) An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65 (6): 570 – 580.

Habbecke M, Kobbelt L (2010) Automatic registration of oblique aerial images with cadastral maps. In: *Proceedings of ECCV Workshop on reconstruction and modeling of large scale 3D virtual environments*

Haberäcker P (1987) *Digitale Bildverarbeitung. Grundlagen und Anwendungen*. Carl Hanser Verlag München Wien, 2nd edition. ISBN 3-446-14901-5.

Hanusch T (2008) A new texture mapping algorithm for photorealistic reconstruction of 3D objects. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 37-5B: 699–706. Proceedings of XXI ISPRS Congress, Beijing.

Hartley RI (1994) Self-calibration from multiple views with a rotating camera. In: Eklundh JO (ed) *Computer Vision - ECCV'94*, volume 800 of *Lecture Notes in Computer Science* (pp. 471–478). Springer Berlin Heidelberg.

Hartley RI, Zisserman A (2004) *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition. ISBN: 978-0521540513.

HEAT (2013) GeoWeb service HEAT. Online: `http://www.saveheat.co/`. Access on June 17th, 2013.

Hebel M, Stilla U (2012) Simultaneous calibration of ALS systems and alignment of multiview LiDAR scans of urban areas. *IEEE Transactions on Geoscience and Remote Sensing*, 50 (6): 2364 – 2379.

Hegarty J, Carswell J (2009) SAMATS - Texture extraction explained. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-5/W1. Proceedings of 3D-ARCH 09 - 3D Virtual Reconstruction and Visualization of Complex Architectures, Trento, Italy.

Hemayed E (2003) A survey of camera self-calibration. In: *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*: 351–357.

Hennigan M (2011) Supply of heat accounts for 47IEA - - largely ignored in climate change debate. Online: `http://www.finfacts.ie/irishfinancenews/article_1022290.shtml`. Access on June 30th, 2011.

Heuel S (2002) *Statistical reasoning in uncertain projective geometry for polyhedral object reconstruction*. PhD thesis, Photogrammetry Department, Institute for Geodesy and Geoinformation, Bonn University, Germany.

Heuel S, Förstner W (2001) Matching, reconstructing and grouping 3D lines from multiple views using uncertain projective geometry. In: *Proceedings of Computer Vision and Pattern Recognition*

Hinz S (2004) *Car detection in aerial thermal imagery by local and global evidence accumulation*. Remote Sensing Technology, Technische Universiäat Müunchen, Technical report.

Hoegner L (2014) *Automatische Texturierung von Gebäudemodellen aus terrestrischen IR-Bildfolgen.* PhD thesis, Fachgebiet Photogrammetrie und Fernerkundung, Ingenieurfakultät für Bau Geo Umwelt, Technische Universität München.

Hoegner L, Iwaszczuk D, Stilla U (2012) Quality assessment for building textures from infrared image sequences. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 39-B3: 391–396. Proceedings of XXII ISPRS Congress, Melbourne, Australia.

Hoegner L, Kumke H, Meng L, Stilla U (2007) Automatic extraction of textures from infrared image sequences and database integration for 3D building models. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2007 (6): 459–468.

Hoegner L, Stilla U (2007) Texture extraction for building models from IR sequences of urban areas. In: *Proceedings of JURSE 2007 - Urban Remote Sensing Joint Event*: 1–6.

Hoegner L, Stilla U (2009) Thermal leakage detection on building façades using infrared textures generated by mobile mapping. In: *Proceedings of JURSE 2009 - Joint Urban Remote Sensing Event*: 1–6.

Höhle J (2008) Photogrammetric measurements in oblique aerial images. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2008 (1): 7–14.

Hough P (1962) Method and means for recognizing complex patterns.

Hsu S, Samarasekera S, Kumar R, Sawhney H (2000) Pose estimation, model refinement, and enhanced visualization using video. In: *Proceedings of Computer Vision and Pattern Recognition*, I: 488–495.

Hu J, You S, Neumann U (2006) Automatic pose recovery for high-quality textures generation. In: *Proceedings of 18th International Conference in Pattern Recognition*: 561–565.

Iwaszczuk D, Avbelj J, Stilla U (2010) Matching von 3D Gebäudemodellen mit Wärmebildern einer flugzeuggetragenen IR-Kamera. In: *30. Wissenschaftlich-Technische Jahrestagung der DGPF*: 334–341.

Iwaszczuk D, Helmholz P, Belton D, Stilla U (2013a) Model-to-Image registration and automatic texture mapping using a video sequence taken by a mini UAV. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 40-1/W1: 151–156. Proceedings of ISPRS Hannover Workshop 2013.

Iwaszczuk D, Hoegner L, Schmitt M, Stilla U (2012a) Hierarchical matching of uncertain building models with oblique airborne IR image sequences. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences* Proceedings of XXII ISPRS Congress, Melbourne, Australia.

Iwaszczuk D, Hoegner L, Schmitt M, Stilla U (2012b) Line based matching of uncertain 3D building models with IR image sequences for precise texture extraction. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2012 (5): 511–521.

Iwaszczuk D, Hoegner L, Schmitt M, Stilla U (2012c) Linienbasiertes Matching von 3D-Gebäudemodellen mit IR Luftbildsequenzen zur automatischen Texturgewinnung. In: *32. Wissenschaftlich-Technische Jahrestagung der DGPF* on CD.

Iwaszczuk D, Hoegner L, Stilla U (2011a) Detection of windows in building textures from airborne and terrestrial infrared image sequences. In: Polish Society for Photogrammetry and Remote Sensing (ed) *Archives of Photogrammetry, Cartography and Remote Sensing*

Iwaszczuk D, Hoegner L, Stilla U (2011b) Detection of windows in IR building textures using masked correlation. In: Stilla U, Rottensteiner F, Mayer H, Jutzi B, Butenuth M (eds) *Proceedings of PIA11 - Photogrammetric Image Analysis*, volume 6952 of *Lecture Notes in Computer Science* (pp. 133–146). Springer.

Iwaszczuk D, Hoegner L, Stilla U (2011c) Matching of 3D building models with IR images for texture extraction. In: Stilla U, Gamba P, Juergens C, Maktav D (eds) *Proceedings of JURSE 2011 - Joint Urban Remote Sensing Event*: 25–28.

Iwaszczuk D, Hoegner L, Stilla U (2013b) Zuordnung von 3D Gebäudemodellen und IR-Videosequenzen mit Linienverfolgung. In: *33. Wissenschaftlich-Technische Jahrestagung der DGPF*: 220–227.

Iwaszczuk D, Hoegner L, Stilla U (2015) Quality-based building-texture selection from different sensors. In: *Proceedings of JURSE 2015 - Joint Urban Remote Sensing Event*: 1–4.

Iwaszczuk D, Stilla U (2010a) A concept for assignment of textures to partially occluded faces of 3D city models stored in CityGML. In: Kolbe T, Koenig G, Nagel C (eds) *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38-4 (W15): 57–62. Proceedings of 3D GeoInfo - International Conference on 3D Geoinformation.

Iwaszczuk D, Stilla U (2010b) Quality measures for textures extracted from airborne IR image sequences. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-3A: 79–84. Proceedings of PCV 2010 - Photogrammetric Computer Vision and Image Analysis, Paris, France.

Iwaszczuk D, Stilla U (2014a) Alignment of 3D building models and TIR video sequences with line Tracking. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2-1. Proceedings of ISPRS Technical Commission I Symposium, Denver, Colorado, USA.

Iwaszczuk D, Stilla U (2014b) Identification of errors in 3D building models by a robust camera pose estimation. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 40-3. Proceedings of Photogrammetric Computer Vision - PCV 2014, ISPRS Technical Commission III Symposium, Zurich, Switzerland.

Iwaszczuk D, Stilla U (2014c) Texturierung von 3D Gebäudemodellen unter Verwendung von Schrägsichtaufnahmen flugzeuggetragener TIR-Kameras. In: *34. Wissenschaftlich-Technische Jahrestagung der DGPF*, 2014.

Jahnke M, Meng L, Kyprianidis JE, Döllner J (2008) Non-photorealistic Rendering on Mobile Devices and its Usability Concerns. In: *Proceedings of Virtual Geographic Environments - An international Conference on Development in Visualization and Virtual Environments in Geographic Information Science* on CD.

Jensen J (2004) *Introductory digital image processing. A remote sensing perspective.* Prentice Hall, 3rd edition. ISBN: 978-0131453616.

Jonst M (2013) Thermalbildauswertung Wien. Verwendungsmoglichkeiten von Thermalbildern im Stadtischen Raum. Master's thesis, Department für Raum, Landschaft und Infrastruktur Institut für Vermessung, Fernerkundung und Landinformation, Universität für Bodenkultur Wien.

Kada M, Klinec D, Haala N (2005) Façade texturing for rendering 3D city models. In: *Proceedings of ASPRS 2005 Annual Conference. "Geospatial Goes Global: From Your Neighborhood to the Whole Planet"*: 78–85.

Kaden R, Kolbe TH (2013) City-wide total energy demand estimation of buildings using semantic 3D city models and statistical data. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-2/W1. Proceedings of the 8th International 3D GeoInfo Conference, Istanbul, Turkey.

Kanatani K (1996) *Statistical optimization for geometric computation: Theory and practice.* Machine intelligence and pattern recognition. Amsterdam, Lausanne, New York: Elsevier. ISBN: 978-0486443089.

Karbo N, Schroth R (2009) Oblique aerial photography: A status review. In: Fritsch D (ed) *Proceedings of Photogrammetric Week 2009*: 119–125.

Karras G, Grammatikopoulos L, Kalisperakis I, Petsa E (2007) Generation of orthoimages and perspective views with automatic visibility checking and texture blending. *Photogrammetric Engineering & Remote Sensing*, 73 (4): 403–411.

Kaul K, Bohn CA (2008) A genetic texture packing algorithm on a Graphical Processing Unit. In: *Proceedings of 11th International Conference on Computer Graphics and Artificial Intelligence*

Kawai S, Saji H (2007) Automatic registration of aerial oblique images and a digital map. In: *Proceedings of SICE, 2007 Annual Conference*: 1791 –1794.

Kirchhof M, Stilla U (2006) Detection of moving objects in airborne thermal videos. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61: 187–196.

Klein AW, Li W, Kazhdan MM, Corrêa WT, Finkelstein A, Funkhouser TA (2000) Non-photorealistic virtual environments. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*: 527–534.

Klinec D (2004) A model based approach for orientation in urban environments. In: *ISPRS International Archives of Photogrammetry and Remote Sensing*, 35-B3: 903–908. Proceedings of XX ISPRS Congress, Commission III, Istanbul, Turkey.

Klingert M (2005) Methoden der digitalen Bildverarbeitung in der Infrarotbauwerksthermografie. In: F. Schley LW (ed) *Forum Bauinformatik*: 287–294.

Koch KR (1999) *Parameter estimation and hypothesis testing in linear models*. Springer, 2nd edition. ISBN: 978-3540652571.

Kolbe T (1999) *Identifikation und Rekonstruktion von Gebäuden in Luftbildern mittels unscharfer Constraints*. PhD thesis, Hochschule Vechta.

Kolbe T, Gröger G, Plümer L (2005) CityGML - interoperable access to 3D city models. In: van Oosterom, Zlatanova F (ed) *Proceedings of International Symposium on Geo-Information for Disaster Management GI4DM, Delft, Netherlands*

Kolecki J, Iwaszczuk D, Stilla U (2010) Calibration of an IR camera system for automatic texturing of 3D building models by direct geo-referenced images. In: *Proceedings of Eurocow 2010*

Kraus K (2007) *Photogrammetry. Geometry from images and laser scans*. Berlin / New York (NY): Walter de Gruyter, 2nd edition. ISBN: 978-3110190076.

Kumke H, Hoegner L, Meng L, Stilla U (2006) Zur automatischen Integration thermischer Texturen und Merkmale in Gebäudemodelle. In: *26. Wissenschaftlich-Technische Jahrestagung der DGPF*: 101–108.

Kuschk G (2013) Model-free dense stereo reconstruction for creating realistic 3D city models. In: *Proceedings of JURSE 2013 - Joint Urban Remote Sensing Event*: 202–205.

Kuzmin Y, Korytnik S, Long O (2004) Polygon-based true orthophoto generation. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 35-B3: 529–531. Proceedings of XX ISPRS Congress, Commission III, Istanbul, Turkey.

Lee S, Jung S, Nevatia R (2002) Automatic pose estimation of complex 3D building models. In: *Proceedings of Workshop on Application of Computer Vision*

Lee S, Nevatia R (2004) Extraction and Integration of Window in a 3D Building Model from Ground View Images. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVP'04*

Legat K (2006) Approximate direct georeferencing in national coordinates. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60 (4): 239–255.

Leibe B, Schiele B (2004) Combined object categorization and segmentation with an implicit shape model. In: *Proceedings of ECCV'04 Workshop on Statistical Learning in Computer Vision*: 1–15.

Lin C, Nevatia R (1995) 3-D descriptions of buildings from an oblique view aerial image. In: *Proceedings of International Symposium on Computer Vision*: 377–382.

Lorenz H (2011) *Texturierung und Visualisierung virtueller 3D-Stadtmodelle*. PhD thesis, Mathematisch-Naturwissenschaftliche Fakultät der Universität Potsdam.

Lorenz H, Döllner J (2006) Façade texture quality estiamation for aerial photography in 3D city models. In: *Proceedings of the Meeting of the ICA Commission on Visualization and Virtual Environments*

Lorenz H, Döllner J (2006) Towards automating the generation of façade textures of virtual city models. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 part2. Proceedings of ISPRS Commission II, WG II/5 Workshop, Vienna, on CD.

Lowe DG (1991) Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (5): 441–450.

Löwner MO, Casper E, Becker T, Benner J, Gröger G, Gruber U, Häfele KH, Kaden, R. Schlüter S (2012) CityGML 2.0 - Ein internationaler Standard für 3D-Stadtmodelle. Teil 2: CityGML in der Praxis. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement (zfv)*, 138 (2): 340–349.

Luhmann T, Ohn J, Piechel J, T. R (2011) Geometrische Kalibrierung von Thermografiekameras. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2011 (1): 5–15.

Luhmann T, Robson S, Kyle S, Harley I (2006) *Close range photogrammetry: Principles, techniques and applications.* Wiley. ISBN 1-870325-50-8.

Luxen M, Förstner W (2001) Optimal camera orientation from points and straight lines. In: Radig B, Florczyk S (eds) *Proceedings of the DAGM 2001*: 84–91.

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

Mayer H, Reznik S (2008) Implicit shape models, model selection, and plane sweeping for 3D facade interpretation. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2008 (3): 187–196.

McGlone J, Mikhail E, Bethel J, Mullen R, for Photogrammetry AS, Sensing R (2004) *Manual of photogrammetry.* American Society for Photogrammetry and Remote Sensing. ISBN: 978-1570830716.

Meidow J, Beder C, Förstner W (2009a) Reasoning with uncertain points, straight lines, and straight line segments in 2D. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64 (2): 125–139.

Meidow J, Förstner W, Beder C (2009b) Optimal parameter estimation with homogeneous entities and arbitrary constraints. In: Denzler J, Notni G (eds) *Pattern Recognition. Proceedings of Symposium of DAGM*: 292–301.

Meier F, Scherer D, Richters J, Christen A (2011) Atmospheric correction of thermal-infrared imagery of the 3D urban environment acquired in oblique viewing geometry. *Atmospheric Measurement Techniques*, 4: 909–922.

Meierhold N, Bienert A, Schmich A (2008) Linebased referencing between images and laser scanner data for image-based point cloud interpretation in a CAD environment. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37-B5: 437–444.

Meixner P, Leberl F (2010) Describing Buildings by 3-Dimensional Details Found in Aerial Photography. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38-A7: 151–156. Proceedings of ISPRS TC VII Symposium 100 Years ISPRS, Vienna, Austria.

Meixner P, Leberl F (2011) Detecting Occlusions in Facade Interpretation from Vertical Aerial Images. In: *Proceedings of Annual Convention of the American Society of Photogrammetry and Remote Sensing, Milwaukee* on CD.

Meng L, Stilla U (2007) Enrichment and multi-purpose visualization of building models with emphasis on thermal infrared data. Online: `http://gepris.dfg.de/gepris/projekt/62030942`. DFG project. Access on October 2nd, 2015.

Michaelsen E (2004) Perceptual grouping using Gestalt laws: Coding as production system, running with data-driven control. In: Ilg U, Bülthoff H, Mallot A (eds) *Proceedings of Dynamic Perception Workshop, Berlin*: 163–168.

Michaelsen E, Iwaszczuk D, Sirmacek B, Hoegner L, Stilla U (2012) Gestalt grouping on facade textures from IR image sequences: Comparing different production systems. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39-B3: 303–308. Proceedings of XXII ISPRS Congress, Melbourne, Australia.

Mikeleit V, Kraneis D (2010) Planen in fünf Dimensionen. *GIS Business*, 2010 (08): 58–59.

Minkina W, Dudzik S (2009) *Infrared thermography: Errors and uncertainties.* Chichester, Wiley. ISBN 978-0-470-74718-6.

MVTec Software GmbH (2014) HALCON operator reference. Online: `http://www.halcon.com/download/reference/index.html`. Access on June 27th, 2014.

Niemeier W (2008) *Ausgleichungsrechnung. Statistische Auswertemethoden.* Berlin/New York: De Gruyter, 2nd edition. ISBN 978-3-11-019055-7.

Nyaruhuma AP, M G, Vosselman G (2012) Verification of 3D building models using mutual information in airborne oblique images. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences.*, I-3: 275–280. Proceedings of XXII ISPRS Congress, Melbourne, Australia.

Ok A, Wegner J, Heipke C, Rottensteiner F, Sörgel U, Toprak V (2012) Matching of straight line segments from aerial stereo images of urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 74: 133–152.

Panday US, Gerke M (2011) Fitting of parametric building models to oblique aerial images. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38-4/W19: 233–238. Proceedings of ISPRS Hannover Workshop 2011: High-Resolution Earth Imaging for Geospatial Information.

Parker J (2011) *Algorithms for image processing and computer vision.* Wiley, 2nd edition. ISBN: 978-0470643853.

Pelagotti A, Del Mastio A, Uccheddu F, F. R (2009) Automated multispectral texture mapping of 3D models. In: *Proceedings of 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland*: 1215–1219.

Phan LN (2012) *Automated rapid thermal imaging systems technology.* PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology. Institute Archives - Noncirculating Collection 3 — Thesis M.E. 2012 Ph.D.

Pollefeys M, Koch R, Van Gool L (1998) Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: *Proceedings of 6th International Conference on Computer Vision*: 90–95.

Remondino F, El-Hakim S (2006) Image-based 3D modelling: A review. *The Photogrammetric Record*, 21 (115): 269–291.

Remondino F, Fraser C (2006) Digital camera calibration methods: Considerations and comparisons. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 part 5: 266–272. Proceedings of ISPRS Commission V Symposium "Image Engineering and Vision Metrology".

Reznik S, Mayer H (2007) Implicit shape models, model selection, and plane sweeping for 3D facade interpretation. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 6: 173–178.

Ripperda N (2008) Grammar based façade reconstruction using RjMCMC. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2008 (2): 83–92.

Ripperda N (2010) *Rekonstruktion von Fassadenstrukturen mittels formaler Grammatiken und Reversible Jump Markov Chain Monte Carlo Sampling.* PhD thesis, Fachrichtung Geodäsie und Geoinformatik, Leibniz Universität Hannover.

Ripperda N, Brenner C (2006) Data driven rule proposal for grammar based façade reconstruction. In: *Pattern Recognition. Proceedings of 28th DAGM Symposium*: 750–759.

Roberts K (1988) A new representation for lines. In: *IEEE Proceedings of Computer Vision and Pattern Recognition*: 635–640.

Sawhney HS, Arpa A, Kumar R, Samarasekera S, Aggarwal M, Hsu S, Nister, D. Hanna K (2002) Video flashlights: real time rendering of multiple videos for immersive model visualization. In: *Proceedings of the 13th Eurographics workshop on Rendering*: 157–168. Pisa, Italy.

Schack L, Schunert A, Soergel U (2012) Lattice detection in persistent scatterer point clouds and oblique aerial imagery. In: *Proceedings of IGARSS - IEEE International Geoscience and Remote Sensing Symposium, 2012*: 451–454.

Schenk T (2004) From point-based to feature-based aerial triangulation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58: 315–329.

Schickler W (1992) Feature matching for outer orientation of single images using 3-D wireframe controlpoints. In: *International Archives for Photogrammetry*, 29-B3: 591–598. Proceedings of ISPRS XVII Congress, Commission III, Washington, DC, USA.

Schneider J, Schindler F, Läbe T, Förstner W (2012) Bundle Adjustment for Multi-camera Systems with Points at Infinity. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3: 75–80. Proceedings of XXII ISPRS Congress, Melbourne, Australia.

Schwermann R (1995) *Geradengestützte Bildorientierung in der Nahbereichsphotogrammetrie.* PhD thesis, Geodädisches Institut der Rheinisch-Westfälische Technischen Hochschule Aachen.

Sester M, Förstner W (1989) Object location based on uncertain models. In: Burkhardt H, Hähne K, Neumann B (eds) *Muster Erkennung 1989, 11th DAGM-Symposium*: 457–464. ISBN:3-540-51748-0.

Sirmacek B, Hoegner L, Stilla U (2011) Detection of windows and doors from thermal images by grouping geometrical features. In: Stilla U, Gamba P, Juergens C, Maktav D (eds) *Proceedings of JURSE 2011 - Joint Urban Remote Sensing Event*: 133–136.

Skaloud J, Legat K (2008) Theory and reality of direct geo-referencing in national coordinates. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63 (2): 272–282.

SketchUp (2015) The easiest way to draw in 3D. Online: http://www.sketchup.com. Access on September 11th, 2015.

Song Y, Shan J (2004) Photorealistic building modeling and visualization in 3-D geospatial information system. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35-B8: 16–20.

States of Jersey (2013) Jersey heat loss map. Online: http://gis.digimap.je/jerseyheatlossmap/. Access on June 25th, 2013.

Stilla U (1993) *Verfahrensvergleich zur automatischen Erkennung in Metall geschlagener Zeichen.* PhD thesis, Fakultät für Elektrotechnik der Universität Fridericiana Karlsruhe.

Stilla U, Kolecki J, Hoegner L (2009) Texture mapping of 3D building models with oblique direct geo-referenced airborne IR image sequences. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Sciences*, 38-1-4-7/W5. Proceedings of Hannover Workshop 2009: High-Resolution Earth Imaging for Geospatial Information, on CD.

Stilla U, Michaelsen E (2002) Detektion von Fahrzeugen in Videosequenzen von luftgetragenen thermischen Kameras. *Österreichische Zeitschrift fr̈ Vermessung und Geoinformation VGI*, 2002 (3+4).

Stilla U, Sörgel U, Jäger K (2000) Generation of 3D-city models and their utilisation in image sequences. In: *ISPRS International Archives of Photogrammetry and Remote Sensing*, 33-B2: 518–524. Proceedings of XIX ISPRS Congress, Commission II, Amsterdam, The Netherlands.

Suveg I, Vosselman G (2004) Reconstruction of 3D building models from aerial images and maps. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58 (3-4): 202–224.

Tang CY, Wu YL, Hu PC, Lin HC, Chen WC (2007) Self-calibration for metric 3D reconstruction using homography. In: *Proceedings of MVA'07*: 86–89.

Technische Universität München (2013) Enrichment and multi purpose visualization of building models with emphasis on thermal infrared data. Project webpage. Online: http://www.pf.bgu.tum.de/prj/blog.html. Access on June 17th, 2013.

ThermalMapper (2013) Project webpage. Online: http://www.faculty.jacobs-university.de/anuechter/thermalmapper.html. Access on June 17th, 2013.

Tian Y, Gerke M, Vosselman G, Zhu Q (2008) Automatic edge matching across an image sequence based on reliable points. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 (B3b): 657–662.

Triggs B, McLauchlan P, Hartley R, Fitzgibbon A (2000) Bundle adjustment – a modern synthesis. In: *Vision algorithms: Theory and practice*, 1883: 298–375.

Tuttas S, Stilla U (2011) Window detection in sparse point clouds using indoor points. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38-3/W22: 131–136. Proceedings of PIA11 - Photogrammetric Image Analysis.

Čech J, Šára R (2008) Windowpane detection based on maximum aposteriori probability labeling. In: Barneva RP, Brimkov V (eds) *Image Analysis - From Theory to Applications. Proceedings of the 12th International Workshop on Combinatorial Image Analysis (IWCIA'08)*: 3–11.

Čech J, Šára R (2009) Languages for constrained binary segmentation based on maximum a posteriori probability labeling. *International Journal of Imaging Systems and Technology*, 19 (2): 69–79.

UVM systems (2015) Corporate website. Online: `http://www.uvmsystems.com`. Access on October 2nd, 2015.

Vallet B, Houzay E (2011) Fast and accurate visibility computation in urban scenes. In: *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*: 77–82. Proceedings of PIA11 - Photogrammetric Image Analysis.

Van den Heuvel F (1998) Vanishing point detection for architectural photogrammetry. In: Chikatsu H, Shimizu E (eds) *ISPRS International Archives of Photogrammetry and Remote Sensing*, 23 part 5: 652–659. Proceedings of ECCV.

Vidas S, Moghadam P, Bosse M (2013) 3D thermal mapping of building interiors using an RGB-D and thermal camera. In: *Proceedings of IEEE International Conference on Robotics and Automation*

virtualcitySYSTEMS (2015) BuildingReconstruction. Online: `http://www.virtualcitysystems.de`. Access on September 11th, 2015.

Vollmer M, Möllmann KP (2010) *Infrared thermal imaging: Fundamentals, research and applications*. New York: Wiley. ISBN: 978-3527407170.

Vosselman G (1992) *Relational matching*. Lecture Notes in Computer Science. Springer. ISBN 978-3-540-55798-2.

Vosselman G (1998) Interactive alignment of parameterised object models to images. In: *ISPRS International Archives of Photogrammetry and Remote Sensing*, 32 (3/1): 272–278.

Wallrabe A (2001) *Nachtsichttechnik. Infrorot-Sensorik physikalische Grundlagen, Aufbau, Konstruktion und Anwendung von Wärmebildgeräten*. Vieweg. ISBN 978-3-663-08100-5.

Wang Y, Schultz S, Guiuffrida F (2008) Pictometry's proprietary airborne digital imaging system and its application in 3D city modelling. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37-B1: 1065–1066. Proceedings of XXI ISPRS Congress, Commission I, Beijing, China.

Weber H (1982) *Thermografie im Bauwesen*. Grafenau: Expert Verlag. ISBN 3885087677.

Weinhaus FM, Devarajan V (1997) Texture mapping 3D models of real-world scenes. *ACM Comput. Surv.*, 29 (4): 325–365.

Weng Q (2009) Thermal infrared remote sensing for urban climate and environmental studies: methods, applications, and trends. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64 (4): 335–344.

Wenzel S, Drauschke M, Förstner W (2007) Detection and description of repeated structures in rectified facade images. *PFG Photogrammetrie Fernerkundung Geoinformation*, 2007 (7): 481–490.

Wenzel S, Drauschke M, Förstner W (2008) Detection of repeated structures in facade images. *Pattern Recognition and Image Analysis*, 18 (3): 406–411.

Wenzel S, Förstner W (2012) Learning a compositional representation for facade object categorization. In: *ISPRS Annals of Photogrammetry, Remote Sensing and the Spatial Information Sciences*, I-3: 197–202. Proceedings of XXII ISPRS Congress, Melbourne, Australia.

Werner T, Zisserman A (2002) New techniques for automated architectural reconstruction from photographs. In: *Proceedings of 7th European Conference on Computer Vision*, II: 541–555.

Wikipedia (2015) Infrared window. Online: `http://en.wikipedia.org/wiki/Infrared_window`. Access on February 12th, 2015.

Yang MY (2011) *Hierarchical and spatial structures for interpreting images of man-made scenes using graphical models.* PhD thesis, Institute of Photogrammetry, University of Bonn.

Yastikli N, Jacobsen K (2005) Direct sensor orientation for large scale mapping – potential, problems, solutions. *The Photogrammetric Record*, 20 (111): 274–284.

Zhang Y, Zhang Z, Zhang J, Wu J (2005) 3D building modelling with digital map, LiDAR Data and video image sequences. *The Photogrammetric Record*, 20 (111): 285–302.

Zlatanova S, Rahman AA, Shi W (2004) Topological models and frameworks for 3D spatial objects. *Computers & Geosciences*, 30 (4): 419–428.

Notation

Font	Example	Description
calligraphic letters	$\mathcal{X}$, $\mathcal{L}$	objects, such as points, lines, planes etc.
italic letters	x, y, z	Euclidean coordinates
upright letters	x, l	homogeneous coordinates
bold letters	$\boldsymbol{x}$, $\mathbf{x}$	vectors, in italic style for Euclidean vectors and upright style for homogeneous vectors
sans-serif letters	R, H	matrices, in italic style for Euclidean matrices and upright style for homogeneous matrices
blackboard bold letters	$\mathbb{A}$	sets

Symbol	Description
$(\,\cdot\,)^{\mathsf{T}}$	transpose
$(\,\cdot\,)^{-1}$	inverse
$(\,\cdot\,)^{-\mathsf{T}}$	transpose of inverse
$\wedge$	join operator ('wedge')
$\cap$	intersection or meet operator ('cap')
$\times$	cross product
$\leftrightarrow$	correspondence
$\mathrm{N_e}(\,\cdot\,)$	Euclidean normalization
$\mathrm{N_s}(\,\cdot\,)$	spherical normalization
I	n×n identity matrix
$0_{m,n}$	m×n zero matrix
$\|\boldsymbol{x}\|$	norm of vector $\boldsymbol{x}$
E	expectation

List of Terms

Term	Explanation
blue spike	infrared radiation with wavelength 3-4.2μm
camera coordinate system	3D Euclidean coordinate system with the origin in the projection center
camera pose	exterior orientation of the camera
co-registration	transformation of two datasets in one coordinate system
direct geo-referencing	geo-referencing based on navigation data
geo-referencing	aligning image data to a world coordinate system
face	a single 3D polygon which is part of a 3D building model
façade structures	structures that can be recognized in the building façade including doors or cornice; in this thesis also thermal leakages
frame	a single image from an image sequence (video)
frame rate	the frequency of delivering a frame by a camera
long wave infrared (LWIR)	infrared radiation with wavelength 7-14μm
red spike	infrared radiation with wavelength 4.2-5μm
texture	a raster assigned to a surface of a 3D model
texture extraction	selection and assignment of image regions to model polygons
thermal infrared (TIR)	infrared radiation with wavelength 3-14μm
thermal leakage	a weak spot in the building hull with a high heat loss
viewing angle	the angle at which an object is observed by the camera

List of Abbreviations

Abbreviation	Description	Page
2D	two-dimensional	19
3D	three-dimensional	9
ALS	airborne laser scanning	19
ARTIST	Automated Rapid Thermal Imaging Systems Technology	14
BIM	Building Information Models	9
B-rep	Boundary Representation	61
CityGML	City Geography Markup Language	21
CSG	Constructive Solid Geometry	61
DGPS	differential global positioning system	18
DLT	direct linear transformation	21
DTM	digital terrain model	13
eTRIMS	E-Training for Interpreting Images of Man-Made Scenes	25
ExtOri	exterior orientation of the camera	16
GIS	geographic information system	10
GPS	global positioning system	18
GPU	Graphics processing unit	23
GSD	ground sample distance	71
HEAT	Heat Energy Assessment Technologies	14
HRSC	High Resolution Stereo Camera	15
ID	identifier	96
INS	inertial navigation system	18
IR3D	a project at Fontazione Bruno Kessler dedicated to thermal building inspections	13
LiDAR	Light Detection And Ranging	25
LoD	Level of Detail	16
rjMCMC	reversible jump Markov chain Monte Carlo	25
ML3DImage	Multi Layer 3DImage	23
RANSAC	RANdom SAmple Consensus	21
RGB	red green blue	13
ROI	region of interest	80
SAR	Synthetic Aperture Radar	15
TIR	thermal infrared	9
TUM	Technische Universität München	13
SAMATS	Semi-Automated Modeling and Texturing System	15
UAV	unmanned aerial vehicle	9
VIS	visible	14

List of Figures

List of Tables

Curriculum Vitae

Name	Dorota Iwaszczuk
Day of birth	14.01.1984
Place of birth	Nisko, Poland
Place of residence	Munich, Germany

Education/Employment

since 2013	Akademische Rätin at Technische Universität München, Astronomical and Physical Geodesy (Munich, Germany)
2009 – 2015	Doctoral Candidate at Technische Universität München, Photogrammetry and Remote Sensing (Munich, Germany) DAAD Research Fellowship Degree: PhD (Doktor Ingenieur)
2009	Engineer at Ingenieurgesellschaft Dr.-Ing. Pecher und Partner (Munich, Germany)
2008 – 2009	Photogrammetrist at GISPRO S.A.(Szczecin, Poland)
2003 – 2008	Studies at AGH, University of Sciences and Technology (Krakow, Poland) Geodesy and Cartography majoring in Geoinformatics, Photogrammetry and Remote Sensing Degree: Master of Science (magister inżynier)
1999 – 2003	General secondary school at Zespół Szkół nr 2 im. Tadeusza Kościuszki (Stalowa Wola, Poland) Mathematics and computer science profile Degree: General qualification for higher education entrance (matura)