Region-Based Pose and Homography Estimation for Central Cameras

Ph.D. Thesis

by

Robert Frohlich

Supervisor:
Prof. Zoltan Kato

External Consultant:
Dr. Levente Tamas

Doctoral School of Computer Science
Institute of Informatics
University of Szeged
Szeged
2019

Contents

List of Algorithms

Acknowledgments

I would like to express first and foremost my sincere gratitude to my advisor, Zoltán Kató, for the continuous support of my Ph.D study and related research. He has been a trustworthy guide both in my research and personal life. Thank you for noticing me and offering this possibility that truly changed my life. I would also like to thank my consultant, Levente Tamás, for his continuous support, good example, and all the insightful comments and encouragement.

I would like to thank Iza for her support and patience during my work. I am grateful to my family for their support and for the opportunities they have created for me to achieve my goals. I am also thankful to my friends and colleagues in the RGVC group.

The research was carried out at the University of Szeged. The work was partially supported by the Doctoral School of the University of Szeged; the NKFI-6 fund through project K120366; "Integrated program for training new generation of scientists in the fields of computer science", EFOP-3.6.3-VEKOP-16-2017-0002; the Ministry of Human Capacities, Hungary through grant 20391-3/2018/FEKUSTRAT; the Research & Development Operational Programme for the project "Modernization and Improvement of Technical Infrastructure for Research and Development of J. Selye University in the Fields of Nanotechnology and Intelligent Space", ITMS 26210120042, co-funded by the European Regional Development Fund; the European Union and the State of Hungary, co-financed by the European Social Fund through projects TAMOP-4.2.4.A/2-11-1-2012-0001 National Excellence Program and FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013); the COST Action TD1201 - COSCH (Colour and Space in Cultural Heritage) through an STSM grant; and finally by the Agence Universitaire de la Francophonie (AUF) and the Romanian Institute for Atomic Physics (IFA), under the AUF-RO project NETASSIST.

Fröhlich Róbert, June 2019.

Chapter 1

Introduction

Computer vision is the scientific field that enables computers to gain high-level understanding from a single digital image or sequence of images. Practically it seeks to automate tasks that the human visual system can do [1]. Main tasks include the acquiring, processing, analyzing and understanding of digital images, and extracting of information about the real world. The countless different applications available today can be enrolled in some well researched sub-domains like scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, segmentation, motion estimation, and image restoration [2]. The methods presented in this thesis propose novel solutions for the 3D pose estimation and planar scene reconstruction problems.

By camera pose in general we can refer to both the absolute and relative pose of cameras. Absolute camera pose estimation consists of determining the position and orientation of a camera with respect to a 3D world coordinate frame, while relative pose refers to the position and orientation with respect to another device (*e.g.* another camera in case of a multi-camera setup), or another position of the same (but moving) camera in a different moment in time. These are fundamental problems in a wide range of applications such as camera calibration, object tracking, simultaneous localization and mapping (SLAM), augmented reality (AR) or structure-from-motion (SfM).

Computer vision methods rely on the image content to estimate the camera's pose. In general, the information retrieved from the image can be of different complexities, starting from points, lines, regions to higher level semantic objects. Using corresponding 2D-3D image points as features to determine the absolute pose is often called the *Perspective-n-Point* (PnP) problem, that can be solved with a minimum number of 3 correspondences [3]. Similarly the *Perspective-n-Line* (PnL) problem, that uses line correspondences in the 2D-3D domain, can also be solved with a minimum number of $n = 3$ feature correspondences.

The methods presented in this thesis rely on patches instead of point or line features, that are higher order, better defined features that bring a few advantages compared to the others. Also the minimum number of corresponding regions needed for pose estimation is $n = 1$. Chapter 2 introduces the reader to the basic aspects of pose and homography estimation. The central spherical camera model is described, which enables us to deal both with traditional perspective and more special dioptric or catadioptric omnidirectional cameras in the same framework. In Chapter 3, first the State-of-the-Art in absolute pose estimation is presented, with an accent on omnidirectional cameras and methods not relying on the classical point features, also reflecting on the possible applications in fields such as the documenting and preserving of cultural heritage objects, and large scale structures, buildings. After this

overview the technical details of the proposed planar region based absolute pose estimation method for spherical cameras are presented, including the special case of perspective cameras and also dealing with non-planar regions. Finally two applications of data fusion are proposed for the documenting of cultural heritage objects and buildings.

The second part of the thesis investigates the possibilities of pose estimation and reconstruction when only 2D information from multiple cameras is available. Focusing again on image regions as features we can define planar homographies acting between the cameras, assuming that these region pairs are the segmented images of the same 3D planar surfaces. Homography estimation is a well researched topic of computer vision and is an essential part of many applications, as such, it can be used to solve different problems including pose estimation or planar reconstruction. In Chapter 4, first the State-of-the-Art in homography estimation is presented, detailing the difficulties involved when working with omnidirectional cameras, and having a particular focus on approaches involving 3D reconstruction. After this overview the first region based homography estimation method proposed for spherical cameras is presented, then two applications related to it, one for relative pose factorization based on these homographies, then a closed form solution for 3D reconstruction with a differential geometric approach. Finally a special homography estimation approach is proposed that can simultaneously provide the relative poses of the cameras and the 3D reconstruction of the planar region(s) in a multi-camera setup. Chapter 5 wraps up the presented results with the main conclusions of this thesis.

Chapter 2

Fundamentals

2.1 Central Omnidirectional Cameras

An omnidirectional (sometimes referred to as panoramic) camera is a camera with a visual field that covers approximately a hemisphere, or the entire hemisphere giving a 360° field of view. There are different ways to build such a camera, either by using a shaped lens (dioptric), using a shaped mirror combined with a standard camera (catadioptric), or using multiple cameras with overlapping field of view (polydioptric). Catadioptric cameras were first used for localizing robots in the early '90s [4] and that is still a major application field for them due to the 360° horizontal field of view. Dioptric, more commonly called *fisheye* cameras started to spread only 10 years later when the manufacturing processes enabled obtaining up to 180° field of view. These cameras' geometry cannot be described using the conventional pinhole model because of the high distortion, thus special models were developed to work with them. In this section two models are presented for central omnidirectional cameras, central meaning that there is a single effective viewpoint, that is the projection center where all optical rays of the viewed objects intersect. Catadioptric cameras can be built to be central using parabolic, hyperbolic or elliptical mirrors [5]. The criteria of single effective viewpoint is important, because it enables the mapping of omnidirectional images onto an image plane forming a planar perspective image, and also enables the use of epipolar geometry. Further more the image can be mapped on a unit sphere centered on the single viewpoint. This spherical projection stands at the base of the two models described in this section, both defining the projection of the camera through a spherical projection of 3D world points that are then mapped to image pixels by some function Φ as shown on the generic model in Fig. 2.1.

The first unified model for central catadioptric cameras was proposed by Geyer and Daniilidis [6] in 2000, which represents these cameras as a projection onto the surface of a unit sphere $\mathcal{S}$ (see Fig. 2.1). According to [6], all central catadioptric cameras can be modeled by a unit sphere, such that the projection of 3D points can be performed in two steps: 1) the 3D point $\mathbf{X}$ is projected onto the unit sphere $\mathcal{S}$, obtaining the intersection $\mathbf{X}_\mathcal{S}$ of the sphere and the ray joining its center and $\mathbf{X}$ (see Fig. 2.1). 2) The spherical point $\mathbf{X}_\mathcal{S}$ is then mapped into the image plane $\mathcal{I}$ through the camera's internal projection function Φ yielding the image $\mathbf{x}$ of $\mathbf{X}$ in the omnidirectional camera. Thus a 3D point $\mathbf{X} \in \mathbb{R}^3$ in the camera coordinate system is projected onto $\mathcal{S}$ by central projection yielding the following

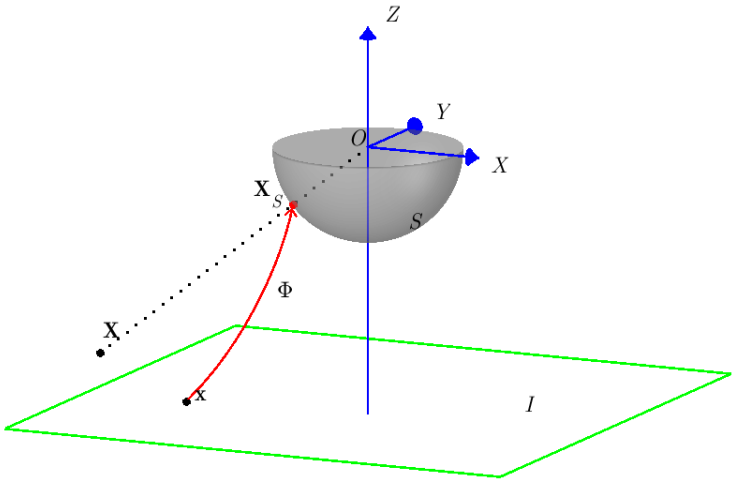

Figure 2.1. A generic spherical camera model.

relation between $\mathbf{X}$ and its image $\mathbf{x}$ in the omnidirectional camera:

$$\Phi(\mathbf{x}) = \mathbf{X}_{\mathcal{S}} = \frac{\mathbf{X}}{\|\mathbf{X}\|} \tag{2.1}$$

This formalism has been widely adopted and various models for the internal projection function Φ have been proposed by many researchers, *e.g.* Micusik [7], Puig [8], Scaramuzza [9] and Sturm [10].

Herein, we will briefly overview two models that have become standards in omnidirectional vision: first the classical specific model of Geyer and Daniilidis [6] for catadioptric cameras, that is not valid for fisheye cameras as shown by [11], then the generic model of Scaramuzza [9] also known as Taylor model, who derived a general polynomial form of the internal projection valid for any type of omnidirectional camera (catadioptric and dioptric as well).

2.1.1 The General Catadioptric Camera Model

Let us first see the relationship between a 3D point $\mathbf{X} = [X_1, X_2, X_3]^T$ and its projection $\mathbf{x}$ in the omnidirectional image $\mathcal{I}$ (see Fig. 2.2). The camera coordinate system is in $\mathcal{S}$, the origin (which is also the center of the sphere) is the *effective projection center* of the camera and the Z axis is the optical axis of the camera which intersects the image plane in the *principal point*. We assume that the axis of symmetry of the mirror is aligned with the optical axis, and X and Y axes of the camera and mirror are also aligned, thus the two coordinate systems only have a translation along Z. To represent the nonlinear (but symmetric) distortion of central catadioptric cameras, Geyer and Daniilidis [6] projects a 3D point $\mathbf{X}$ from the camera coordinate system to omni image pixel $\mathbf{x}$ through four steps. First $\mathbf{X}$ is centrally projected onto the unit sphere:

$$\mathbf{X}_S = \frac{\mathbf{X}}{\|\mathbf{X}\|} = (X_S, Y_S, Z_S)$$

Then point coordinates are changed to a new reference frame centered in $\mathcal{C}_\xi = (0, 0, -\xi)$:

$$\mathbf{X}_\xi = (X_S, Y_S, Z_S + \xi)$$

where ξ ranges between 0 and 1. $\mathbf{X}_\xi$ is then projected onto the normalized image plane:

$$\mathbf{m} = (x_m, y_m, 1) = (\frac{X_S}{Z_S + \xi}, \frac{Y_S}{Z_S + \xi}, 1) = \Phi^{-1}(\mathbf{X}_S)$$

In the final step the point $\mathbf{m}$ is mapped to the camera image point $\mathbf{x}$ using the camera calibration matrix $\mathbf{K}$

$$\mathbf{x} = \mathbf{K}\mathbf{m}$$

where

$$\mathbf{K} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

contains the camera's focal length and optical center coordinates.

This model was later refined by Barreto and Araujo [12], where they considered that oriented projective rays $\mathbf{x}_c$ are mapped for each 3D point $\mathbf{X}_w = [X_1, X_2, X_3, 1]^T$ expressed in world coordinate system with homogeneous coordinates ($\mathbf{x}_c = \mathbf{T}\mathbf{X}_w$, where $\mathbf{T}$ is a rigid body transformation), and their corresponding projective rays $\mathbf{x}_{cam}$ intersect in the mirror surface

$$\mathbf{x}_{cam} = \mathbf{M}_c h(\mathbf{x}_c)$$

where $\mathbf{M}_c$ includes the mirror parameters ξ and ψ (see [12] for details) and $h(\mathbf{x}_c)$ can be interpreted as a non-linear mapping between two oriented projective planes:

$$\mathbf{x}_\mathcal{P} = h(\mathbf{x}_c) = \begin{bmatrix} X_1 \\ X_2 \\ X_3 + \xi\sqrt{X_1^2 + X_2^2 + X_3^2} \end{bmatrix}$$

The virtual plane $\mathcal{P}$ is then transformed in the image plane $\mathcal{I}$ (see Fig. 2.2) through the homography $\mathbf{H}_C$ as

$$\begin{aligned} \mathbf{x} &= \mathbf{H}_C \mathbf{x}_\mathcal{P} = \mathbf{H}_C h(\mathbf{x}_c) \\ \mathbf{H}_C &= \mathbf{K}_C \mathbf{R}_{CM} \mathbf{M}_C, \end{aligned}$$

where $\mathbf{K}_C$ includes the perspective camera parameters (taking the picture of the mirror), $\mathbf{R}_{CM}$ is the rotation between camera and mirror. Thus the relation between image point $\mathbf{x}$ and rays $\mathbf{x}_{cam}$ is given by a collineation depending on camera orientation and internal parameters. Herein, we will assume an ideal setting: no rotation (*i.e.* $\mathbf{R}_{CM} = \mathbf{I}$) and a simple pinhole camera with focal length f and principal point (x_0, y_0) yielding

$$\mathbf{H}_C = \begin{bmatrix} f(\psi - \xi) & 0 & x_0 \\ 0 & f(\xi - \psi) & y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \gamma & 0 & x_0 \\ 0 & -\gamma & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where γ is the generalized focal length of the camera-mirror system.

According to [6] and [12], this representation includes:

1. catadioptric systems containing a hyperbolic mirror and a perspective camera for $0 < \xi < 1$, as well as

2. catadioptric systems with parabolic mirror and orthographic camera for $\xi = 1$ and

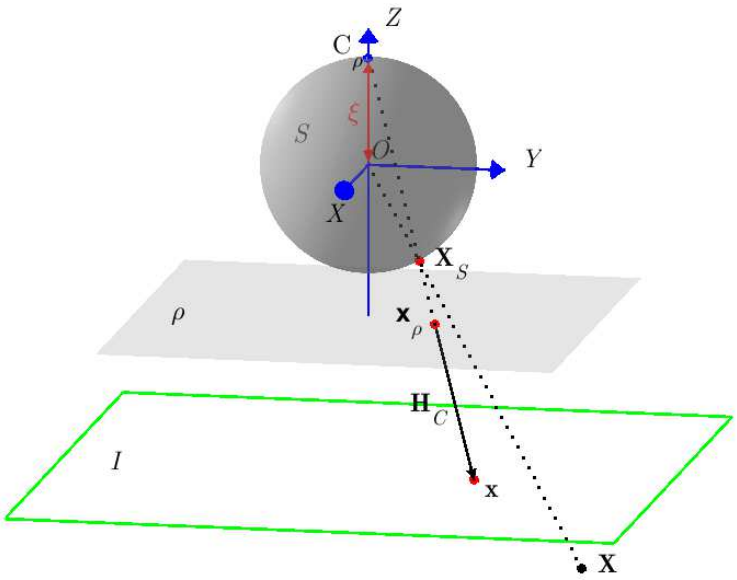

Figure 2.2. Omnidirectional camera model using Geyer and Daniilidis' representation [6].

3. conventional perspective cameras as $\xi = 0$

In the following, without loss of generality, we will focus on case 2). The bijective mapping $\Phi : \mathcal{I} \rightarrow \mathcal{S}$ is the inverse of the camera's projection function, which is composed of 1) transforming the image point $\mathbf{x} \in \mathcal{I}$ back to the $\mathcal{P}$ virtual projection plane by $\mathbf{H}_C^{-1}$:

$$\mathbf{x}_{\mathcal{P}} = \mathbf{H}_C^{-1}\mathbf{x},$$

and then 2) projecting back this point $(x_{\mathcal{P}1}, x_{\mathcal{P}2}, x_{\mathcal{P}3})^T$ from $\mathcal{P}$ to a 3D ray through the virtual projection center $\mathbf{C}_{\mathcal{P}}$ (assuming $\xi = 1$):

$$\begin{aligned} \mathbf{X} &= h^{-1}(\mathbf{x}_{\mathcal{P}}) = \begin{bmatrix} x_{\mathcal{P}1} \\ x_{\mathcal{P}2} \\ \frac{x_{\mathcal{P}3}^2 - x_{\mathcal{P}1}^2 - x_{\mathcal{P}2}^2}{2x_{\mathcal{P}3}} \end{bmatrix} \\ &= h^{-1}(\mathbf{H}_C^{-1}\mathbf{x}) = \begin{bmatrix} \frac{1}{\gamma}(x_1 - x_0) \\ \frac{1}{\gamma}(x_2 - y_0) \\ \frac{1}{2}\left(1 - \left(\frac{x_1 - x_0}{\gamma}\right)^2 - \left(\frac{x_2 - y_0}{\gamma}\right)^2\right) \end{bmatrix} \end{aligned} \tag{2.2}$$

We thus get the following expression for $\Phi : \mathcal{I} \rightarrow \mathcal{S}$:

$$\Phi(\mathbf{x}) = \mathbf{X}_{\mathcal{S}} = \frac{h^{-1}(\mathbf{H}_C^{-1}\mathbf{x})}{\|h^{-1}(\mathbf{H}_C^{-1}\mathbf{x})\|} \tag{2.3}$$

which provides the corresponding spherical point $\mathbf{X}_{\mathcal{S}} \in \mathcal{S}$. $\nabla\Phi$ is easily obtained from (2.2) and (2.3).

2.1.2 Scaramuzza's Omnidirectional Camera Model

The model presented by Micusik and Pajdla [13] also relied on the same idea of projecting on the unit sphere, but used a different parametrization for the projection function. It was

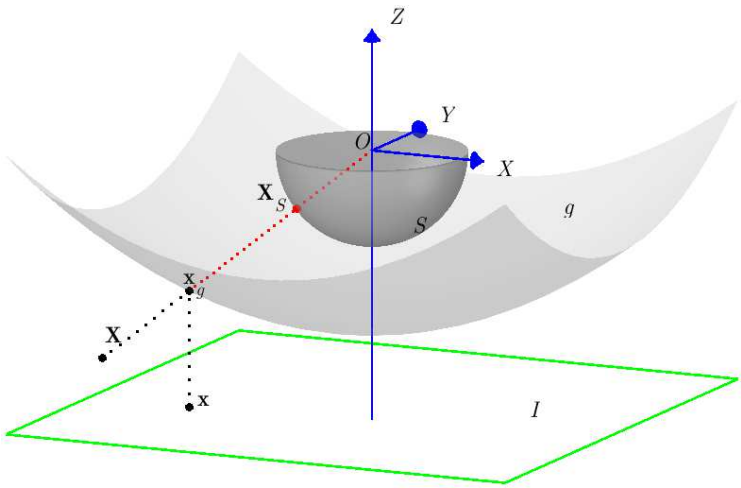

Figure 2.3. Omnidirectional camera model using Scaramuzza's representation [9, 14].

already capable of handling both catadioptric and fisheye cameras, but unfortunately the parameters of the two functions that describe the projection had to be determined uniquely for every sensor, thus the use of the model was cumbersome. Instead, the general polynomial form proposed by Scaramuzza *et al.* [9, 14] is easier to apply for different types of cameras. Following [9, 14], we assume that the camera coordinate system is in $\mathcal{S}$, the origin is the effective projection center of the omnidirectional camera. Let us first see the relationship between a point $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^2$ in the image $\mathcal{I}$ and its representation $\mathbf{X}_\mathcal{S} = [X_{\mathcal{S}1}, X_{\mathcal{S}2}, X_{\mathcal{S}3}]^\top \in \mathbb{R}^3$ on the unit sphere $\mathcal{S}$ (see Fig. 2.3). Note that only the half sphere on the image plane side is actually used, as the other half is not visible from image points.

There are several well known geometric models for the internal projection [5–7, 9]. To represent the nonlinear (but symmetric) distortion of central omnidirectional optics, [9, 14] places a surface g between the image plane and the unit sphere $\mathcal{S}$, which is rotationally symmetric around z (see Fig. 2.3). The details of the derivation can be found in [9, 14]. As shown by the authors, polynomials of order three or four are suitable for accurately modeling all commercially available catadioptric and many types of fisheye cameras as well, thus we used a fourth order polynomial:

$$g(\|\mathbf{x}\|) = a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4, \tag{2.4}$$

which has 4 parameters (a_0, a_2, a_3, a_4) representing the internal parameters of the camera (only 4 parameters as a_1 is always 0 according to [14]). The bijective mapping $\Phi : \mathcal{I} \to \mathcal{S}$ is composed of

1. lifting the image point $\mathbf{x} \in \mathcal{I}$ onto the g surface by an orthographic projection

$$\mathbf{x}_g = \begin{bmatrix} \mathbf{x} \\ a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4 \end{bmatrix} \tag{2.5}$$

2. then centrally projecting the lifted point $\mathbf{x}_g$ onto the surface of the unit sphere $\mathcal{S}$:

$$\mathbf{X}_\mathcal{S} = \Phi(\mathbf{x}) = \frac{\mathbf{x}_g}{\|\mathbf{x}_g\|} \tag{2.6}$$

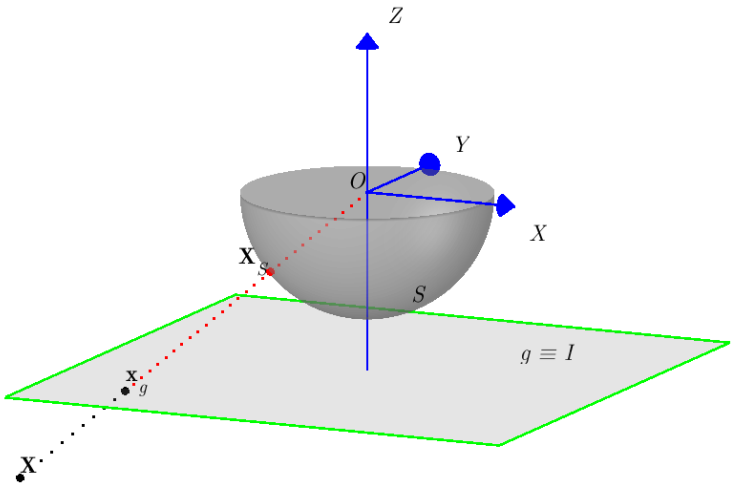

Figure 2.4. Perspective camera model using Scaramuzza's spherical representation, assuming $g \equiv \mathcal{I}$.

Thus the omnidirectional camera projection is fully described by means of unit vectors $\mathbf{X}_{\mathcal{S}}$ in the half space of $\mathbb{R}^3$ and these points correspond to the unit vectors of the projection rays. The gradient of Φ can be obtained from (2.5) and (2.6).

Throughout the works presented in this thesis we used the above described spherical camera model to work with omnidirectional cameras.

Spherical Model of the Perspective Camera

It's clear to see that by introducing the polynomial surface g, the camera model described in Section 2.1.2 can model the different distortion and the large field of view of omnidirectional cameras, solely determined by the parameters (2.4) of the g surface. Consequently if we set all parameters of g to be zero, except the constant a_0, we get a perspective camera (g is a planar surface parallel to the image plane) as seen in Fig. 2.4. Further simplifying the model we can choose $g \equiv \mathcal{I}$ by using $a_0 = f$ (f is the focal length, the distance from the projection center to the image plane), the bijective mapping $\Phi : \mathcal{I} \to \mathcal{S}$ for a perspective camera becomes simply the unit vector of $\mathbf{x}$:

$$\mathbf{X}_{\mathcal{S}} = \Phi(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{2.7}$$

This special case will be discussed later in Chapter 3.2.2.

2.2 Absolute and Relative Pose

Let's consider an arbitrary right handed world coordinate frame $\mathcal{W}$ with a 3D point $\mathbf{X}_{\mathcal{W}}$ in it, given by $\mathbf{X}_{\mathcal{W}} = [X_1, X_2, X_3, 1]^T$ homogeneous coordinates. A camera $\mathcal{C}$ placed in the same space has it's coordinate system chosen as X axis pointing right, Y axis down and Z axis pointing forward in the direction of the optical axis, as shown in Fig. 2.5. The relation between the world coordinate system and the one attached to the camera is given by the absolute camera pose, a rigid body transformation noted as $\mathbf{T} = (\mathbf{R}, \mathbf{t})$, composed

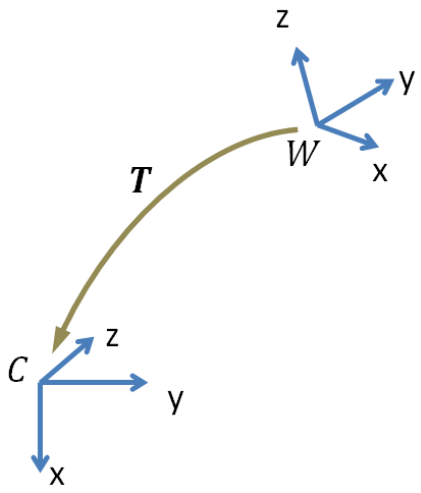

Figure 2.5. Absolute pose of camera $\mathcal{C}$ to the world reference frame $\mathcal{W}$. $\mathbf{T}$ is acting on the points given in $\mathcal{W}$.

of a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ as a 3×4 matrix:

$$\mathbf{T} = [\mathbf{R}|\mathbf{t}] = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \tag{2.8}$$

where r_{ij} are the row-column indexed elements of the rotation matrix, i being the row, j the column index, and t_x, t_y, t_z the components of the translation vector.

By the convention used, the above defined absolute pose is acting on 3D points $\mathbf{X}_{\mathcal{W}}$ given in $\mathcal{W}$, transforming them into the coordinate system of $\mathcal{C}$. Another definition of the absolute pose would describe the transformation that gives the position and orientation of the camera in $\mathcal{W}$, that is basically the inverse transformation of $\mathbf{T}$. This is more widely used in applications where the camera's pose as an object in space is relevant (*e.g.* robotics). Since we are more interested in the projection of the camera then its position in the world, in our work $\mathbf{T}$ described at (2.8) is part of the camera matrix $\mathbf{P}$ used for projection to the image plane, thus it is acting on the 3D points:

$$\mathbf{x} \cong \mathbf{P}\mathbf{X}_{\mathcal{W}} = \mathbf{K}\mathbf{T}\mathbf{X}_{\mathcal{W}} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}_{\mathcal{W}}, \tag{2.9}$$

where '$\cong$' denotes the equivalence of homogeneous coordinates, *i.e.* equality up to a non-zero scale factor, and $\mathbf{K}$ is the 3×3 upper triangular calibration matrix containing the internal projection parameters of the perspective camera. Exactly the same transformation applies if we consider the spherical camera model, as shown at the beginning of Chapter 2.1.1, the 3D point $\mathbf{X}$ expressed in camera coordinate system is practically obtained as $\mathbf{X} = \mathbf{T}\mathbf{X}_{\mathcal{W}}$.

If we consider multiple cameras in the same framework, or if we intend to work on the image sequence of a moving camera the absolute pose of each camera/frame can be defined individually, but depending on the application this might not be always the best solution. In tracking, localization or reconstruction related applications the relative camera pose between neighboring cameras or consecutive frames is often more interesting since it provides vital information about the actual local state of the system, while absolute pose provides a more global information useful for calibration, building a map or navigating to a predefined location.

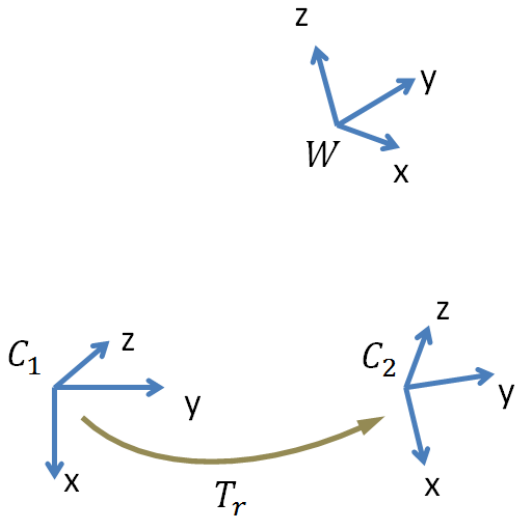

Figure 2.6. Relative pose of two cameras, acting on points in $\mathcal{C}_1$.

By relative pose we refer to the rigid body transformation $\mathbf{T_r}$ that acts between the coordinate systems of two cameras. Practically, compared to the absolute pose, the difference stands in the definition of the global coordinate system. One of the cameras can be assigned the role of the reference frame, and other cameras' absolute pose is expressed relative to that, resulting the relative pose between cameras. In case of an image sequence this can be applied incrementally if needed, each camera taking the role of reference frame for the next frame. By definition $\mathbf{T_r}$ acts on the points expressed in the reference coordinate system, thus in the example shown in Fig. 2.6 the relative pose brings 3D points expressed in $\mathcal{C}_1$ into $\mathcal{C}_2$, thus the $\mathbf{T}_{\mathbf{r}_1}^2$ notation can also be used. The absolute pose of $\mathcal{C}_1$ and $\mathcal{C}_2$ and their relative pose $\mathbf{T}_{\mathbf{r}_1}^2$ satisfy the following relation:

$$\mathbf{T}_2 = \mathbf{T}_{\mathbf{r}_1}^2 \mathbf{T}_1 \tag{2.10}$$

Estimating the Camera Pose

The most standard method for estimating camera pose, the Perspective-n-Point (PnP) problem originates from camera calibration. These methods rely on corresponding data (called features) in the reference world coordinate system and in the camera frame. All PnP problems include the P3P problem as a special case, $n = 3$ being the minimum number of features, for which the problem can be solved. This special case is known to provide up to four solutions that can then be disambiguated using a fourth point. In another special case if the points are coplanar, a homography transformation can be exploited instead [15]. A standard approach for the PnP problem is first using P3P in a RANSAC scheme [16] to remove the outliers, then PnP on all remaining inliers. All the P3P algorithms first estimate the distance of points from the camera center expressing them in the camera coordinate system, then estimating the transformation $\mathbf{T}$ that aligns them to the points expressed in world coordinate system using closed-form solution. Other methods rely on the minimizing of feature projection errors. The possibilities of using different features such as lines, contours, regions, objects are actively researched, but so far the point based features (*e.g.* SIFT, SURF, AKAZE) are most commonly used in applications (a recent comparative analysis of these can be found in [17]). This thesis presents novel region-based registration meth-

ods that do not rely on any point features, nor intensity information, only using segmented planar image regions.

2.3 Planar Homography

In general terms a homography is a non-singular, line preserving, invertible projective mapping from an n dimensional space to itself, represented by a square $(n+1)$ size matrix, having $(n+1)^2-1$ degrees of freedom. In case of 2D planar homographies we have a 3×3 matrix representation with 8 DoF acting between planes defined in 3D space.

Let's assume we have two cameras observing a scene that contains a plane $\pi = (\mathbf{n}^T, d)^T$ so that for points $\mathbf{X}$ on the plane $\mathbf{n}^T\mathbf{X} + d = 0$. For simplicity we can choose the world origin to be in the projection center of one of the cameras, thus plane π and point $\mathbf{X}$ are both defined in the camera coordinate system, then the camera matrices will be:

$$\mathbf{P}_1 = \mathbf{K}_1[\mathbf{I}|\mathbf{0}] \text{ and } \mathbf{P}_2 = \mathbf{K}_2[\mathbf{R}|\mathbf{t}] \tag{2.11}$$

According to [15] a homography induced by the plane π, acting between the normalized image planes of $\mathcal{C}_1 \rightarrow \mathcal{C}_2$ is composed as:

$$\mathbf{H} \propto \mathbf{R} - \mathbf{t}\mathbf{n}^T/d \tag{2.12}$$

More specifically, considering the homography acting between image pixels $\mathbf{x}_1$ of $\mathcal{C}_1$ and $\mathbf{x}_2$ of $\mathcal{C}_2$ we have $\mathbf{x}_2 = \mathbf{H}_{im}\mathbf{x}_1$ where

$$\mathbf{H}_{im} = \mathbf{K}_2(\mathbf{R} - \mathbf{t}\mathbf{n}^T/d)\mathbf{K}_1^{-1} \tag{2.13}$$

but since we are going to work with calibrated cameras, we can consider the calibration matrices are known, thus we can work in the normalized image plane points (unit vectors in case of the spherical camera model), using the notation in (2.12) that basically acts between the projection rays of the points.

Since in most of the applications only the individual camera images are available, the planar homographies have to be computed directly from corresponding image elements that specify the plane. Since $\mathbf{H}$ has 8 DoF (one free scale factor) it is enough to find 4 point matches on the camera images lying on the image of plane π to be able to determine $\mathbf{H}$. Using these four points in a general position (*i.e.* no three of them are collinear) $\mathbf{H}$ can be calculated using the Direct Linear Transform (DLT) algorithm.

In case of known epipolar geometry, if the fundamental matrix is available, $\mathbf{H}$ can be calculated using three non collinear points, or a line and a point that both define a plane uniquely. If the fundamental matrix is not available, it can be computed using the idea of plane induced parallax and 6 image points, 4 coplanar points define the homography, and the two points off the plane provide constraints to determine the epipole [15]. Other solutions rely on conics, curves, discrete contours, or even planar texture [18].

Retrieving Pose and Plane from Homography

According to [15] the knowledge of homographies between the images means that we know the first 3×3 part of the camera matrix $\mathbf{P} = [\mathbf{M}|\mathbf{t}] = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, that in case of calibrated cameras means that the orientation can be estimated from it, (*e.g.* based on vanishing points)

hence only the last column (translation) has to be computed. [19] proposed a method for recovering the relative pose and also the projective shape through SVD factorization of a measurement matrix, using only fundamental matrices and epipoles estimated from the image data.

Relative pose and plane parameters can also be easily retrieved from planar homographies if some special constraints can be applied on the problem. For example we show that considering a weak *Manhattan World* setup with vertical planes in the scene, and the vertical direction of the camera known from external source, we can directly decompose this special homography to find the unknown rotation angle, translation and plane normal. In this thesis we also present a novel homography estimation framework based on planar regions, that enables us to develop a simultaneous relative pose and plane parameter estimating algorithm (based on the parametrization (2.12)), and also a differential geometric approach for plane reconstruction from homographies.

Chapter 3

Absolute Pose Estimation and Data Fusion

3.1 State of the Art Overview

Absolute pose estimation of a camera with respect to a 3D world coordinate frame is a fundamental building block in various computer vision applications, such as robotics (*e.g.* visual odometry [20], localization and navigation [21]), augmented reality [22], geodesy, or cultural heritage [Frohlich *et al.*, 2016]. There is also considerable research effort invested in autonomous car driving projects both on academic and industrial side. While for the specific scenarios such as highways there are already a number of successful applications, this problem is still generally not solved for complex environments such as the urban ones [23, 24]. Recent developments in the autonomous driving, especially in urban environment, are using a great variety of close-to-market sensors including different cameras and active sensors, this puts into focus the need for information fusion emerging from these sensors [25].

The absolute pose estimation problem has been extensively studied yielding various formulations and solutions. Most of the approaches focus on a single perspective camera pose estimation using *n* 2D–3D point correspondences. One of the earliest works to consider this problem was [16] who also coined the term *Perspective-n-Point* (or PnP) for this type of problem with *n* feature points. Later [26] proposed a method based on the iterative improving of the pose computed with a weak perspective camera model, that converges to a pose estimation computed with a perspective camera model, then [27] gave an algebraic derivation of this method. More recently [28] proposed a non-iterative solution that had a linear computational complexity growth with *n*, then [29] proposed the first non-iterative solution (RPnP) that achieved higher accuracy than the iterative State-of-the-Art methods with less computation time. The first Unified PnP (UPnP) solution that unifies all the desirable properties of previous algorithms was proposed by [30]. The PnP problem has been widely studied not just for large *n* but also for the minimal case of $n = 3$ (see [30] for a recent overview). More recently researchers started using line correspondences instead of points, that yields the *Perspective-n-Line* (PnL) problem [32, 31] (see [31] for a detailed overview).

Several applications dealing with multimodal sensors make use of fused 2D radiometric and 3D depth information. The availability of 3D data has also became widespread. 3D measurements of a scene can be provided both by the classical image-based techniques,

such as Structure from Motion (SfM) [33], and modern range sensors (*e.g.* Lidar, ToF) that record 3D structure directly. Therefore methods to estimate absolute pose of a camera based on 2D measurements of the 3D scene are still actively researched [34, 30, 35]. Many of these methods apply to general central cameras (both perspective and omnidirectional) that are often represented by a unit sphere [5–7, 9].

In order to obtain a common coordinate frame for these devices the relative position among the different 2D and 3D cameras has to be determined. Although application specific solutions exist, the principles of the relative position estimation are still similar. The main challenge in the accurate calibration is due to the uncertainty in the relative position measurement among different sensor bases. Fortunately, the calibration of the central cameras including the perspective or omnidirectional ones can be encapsulated in a common theoretical framework. For both types of cameras a clear distinction is made for the intrinsic and extrinsic calibration.

Internal calibration refers to the self parameters of the camera, while external parameters describe the *pose* of the camera with respect to a world coordinate frame. While internal calibration can be solved in a controlled environment, *e.g.* using special calibration patterns, pose estimation must rely on the actual images taken in a real environment. Although non-conventional central cameras like catadioptric or dioptric (*e.g.* fisheye) panoramic cameras have a more complex geometric model, their calibration also involves internal parameters and external pose. Popular methods rely on point correspondences such as [14], or using fiducial markers [36], which may be cumbersome to use in real life situations. This is especially true in a multimodal setting, where images need to be combined with other non-conventional sensors like Lidar scans providing range only data. The Lidar-omnidirectional camera calibration problem was analyzed from different perspectives. Recently, the geometric formulation of omnidirectional systems were extensively studied [7, 37, 38]. The internal calibration of such cameras depends on these geometric models. Although different calibration methods and toolboxes exist [36, 39, 14], this problem is by far not trivial and is still in focus [38]. In [40], the calibration is performed in natural scenes, however point correspondences between the 2D-3D images are selected in a semi-supervised manner. In [41], calibration is tackled as an observability problem using a (planar) fiducial marker as calibration pattern. In [42] a fully automatic method is proposed based on mutual information (MI) between the intensity information from the depth sensor and the omnidirectional camera, while in [44, 43] a deep learning approach for calibration is presented. Another global optimization method uses the gradient orientation measure as described in [45]. However, these methods require range data with recorded intensity values, which are not always available. In real life applications, it is also often desirable to have a flexible one step calibration for systems which do not necessarily contain sensors fixed to a common platform.

3.1.1 Related Work

Due to the large number of applications using central camera systems, also the range of the calibration methods is rather wide. Beside solving the generic 2D-3D registration problem, several derived applications exist including medical [46], robotics [45] and cultural heritage ones [Frohlich *et al.*, 2016]. For the pose estimation in known environment a good example can be found in [47], while in [48] an application is reported using spherical image fusion with spatial data.

A possible differentiation for the applications is related to the input data properties,

such as image resolution. For high precision image registration the work presented in [49] is based on the information of the Lidar scan intensity or the ground elevation level. Mutual information is computed between the two images and fed to a global optimization algorithm in order to estimate the unknown camera parameters. The algorithm proved to be successful in urban environment. For low precision and high frame rate systems such as the ones used for navigation purposes, the registration challenges are addressed in different ways. In these setups several Lidar-camera scan pairs are acquired and the registration is performed for these image pairs as described in [50].

A more generic classification of the types of algorithms is presented in [51]. Beside the direct measured relative pose methods such as [52], a number of generic methods are summarized below.

Feature-based Methods

Several variants for calibration based on specific markers are used for extrinsic [53, 54] camera calibration. In the early work of [55], alignment based on a minimal number of point correspondences is proposed, while in [56], a large number of 2D-3D correspondences are used with possibly redundant or mismatched pairs. The extrinsic calibration of 3D lidar and low resolution perspective color camera was among the first addressed in [57] which generalized the algorithm proposed in [58]. This method is based on manual point feature selection from both domains and assumes a valid camera intrinsic model for calibration. A similar manual point feature correspondence based approach is proposed in [40]. Recently, increasing interest is manifested in various calibration setups ranging from high-resolution spatial data [49] to low-resolution commercial cameras [59]. Also online calibration for different measurements in time such as in case of a moving platforms containing depth and color sensors are presented in [60, 42].

Color-intensity-based Methods

A popular alternative to the feature based matching is the mutual information extraction and alignment between the 2D color and the 3D data with intensity information such as in case of [61, 45]. Extensions to the simultaneous intrinsic-extrinsic calibration are presented in the work [41] which makes use of lidar intensity information to find correspondences between the 2D-3D domains. Other works are based on the fusion of IMU or GPS information in the process of calibration [62].

Statistical Methods

A good overview of the statistical techniques based calibration methods can be found in [46]. Mutual information extraction based on particle filters is presented in the work [45] which performs the calibration based on the whole image space of a single 2D-3D observation. The calibration can be based both on intensity and normal distribution information for the 3D data. A further extension of this approach based on gradient orientation measure is described in [63]. A gradient information extraction and global matching between the 2D color and 3D reflectivity information is presented in [42]. This has two major differences compared to the work described in this paper. The current work is not limited to lidar systems with reflectivity information rather it is based only on depth information. On

the optimization side, the proposed method is not restricted to convex problems and allows camera calibration using only a single Lidar-camera image pair.

Silhouette-based Methods

An early and efficient silhouette based registration method is presented in [64], which solves a model-based vision problem using parametric description of the model. This method can be used with an arbitrary number of parameters describing the object model and is based on a global optimization with the *Levenberg-Marquardt* method. A whole object silhouette based registration is proposed in [61], where the authors describe the 2D-3D registration pipeline including segmentation, pixel level similarity measure and global optimization of the registration. Although the proposed method can be used in an automatic manner, this is limited only to scenes with highly separable foreground-background parts. By an automatic segmentation of the relevant forms in panoramic images, which are registered against cadastral 3D models the segmented regions are aligned using particle swarm optimization in [65]. An extension of the silhouette based registration method is proposed in [66]. In this work a hybrid silhouette and keypoint driven approach is used for the registration of 2D and 3D information. The advantage of this method is the possibility of multiple image registration as well as the precise output of the algorithm.

3.1.2 Cultural Heritage Applications

From a cultural heritage application's point of view there are completely different criteria that have to be considered, primarily the availability of the devices and the measurement method that they require are key aspects. Recently, as more and more 3D imaging devices and methods are available, cultural heritage experts have a several options to choose from for documenting architectures, excavation sites, caves [67], historical scenes or other large or small scale objects. Thus the need for effective software solutions is also increasing. Capturing an object with different modalities giving different levels of detail, the fusion of these data is inevitable at a given point. Different devices working on different principles impose a specific workflow for the creation of a colorized 3D model. But unfortunately, as it is well known for all experts working in this field, there is no one single solution that could be used for all types of case studies.

In archaeological cultural heritage study 3D modeling has become a very useful process to obtain indispensable data for 3D documentation and visualization. While the precise surveying and measurement of architecture, or excavation sites is possible with total stations (*e.g.* manufactured by Leica Geosystems), the use of these devices and the creation of a precise model based on the measurements needs highly experienced professionals. Using a Lidar scanner instead, one can also produce a metric 3D model, with relatively high precision, that could be sufficient for most tasks, and could be used for completing different measurements later on the data itself, even special measurements impossible to perform in real world. As we found out it can also be indispensable for planning the renovation process of some cultural heritage buildings that were never measured properly before, since the plans can only be designed once a complete model of the building's actual state is available. Also spatial and color features are important factors for specialists to analyze the ruins of some historical building, make hypothesis about the 3D models and obtain a 3D view of the assumed original look of the structure, to use it then as an educational or research tool.

Another important cultural heritage application is the creation of accurate 3D models of small scale objects, like ceramics or fragments, including textural details. This represents a better, new way of documenting ceramics next to the traditional 2D representations through technical drawings. Beyond the accuracy of the 3D features such as structural surfaces and shapes, archeologists are also concerned by the accuracy of color features, especially color patterns and color inclusions. Indeed spatial and color features are important factors for specialists in ceramics to analyze fragments, make hypothesis about 3D objects/shapes from sets of fragments, and in general as educational and research tools.

Reviewing recent cultural heritage publications we can observe, that based on the actual case's properties and the available budget, different groups used completely different approaches starting from the low cost options like photogrammetry or relatively cheap, entry level structured light scanners up to the more professional Lidar scanners and even high-end, expensive laboratory setups producing the best possible results. As strict laboratory conditions can hardly be ensured on the field, and not all case studies require the highest possible precision of the results, usually some compromises are made as long as the quality of the results still meets the project's needs.

Most of the recent works rely on either laser, structured light based 3D scanners, photogrammetry or a combination of these to obtain the 3D model of an object. Though photogrammetry is widely used, recent overviews of available techniques presented in [69, 68] show its main disadvantages: a large number of images has to be captured without any feedback, not being able to verify partial results on the go, and processed later on powerful workstations that is also time consuming. The level of detail captured can only be verified after the final reconstruction is finished, if accidentally some parts were not captured from enough viewpoints, it can only be corrected by a new acquisition. In order to overcome this issue, the authors of [70] have experimented with a mathematical positioning procedure to reduce the required number of images captured and ensure a high level of detail over all regions. Others use various software solutions to do the 3D reconstruction using more images taken from arbitrary positions [71]. Since most commercial software rely on the detection of some keypoints, problems can occur with objects having no texture at all. In these cases, the best practice is placing external markers near or on the object if it is possible, visible on the captured images. A good example is presented in [72], where geotagged marker points were used for both photogrammetric and laser scanning techniques.

3D scanners on the other hand are generally more expensive devices than the DSLR cameras used for photogrammetry, but they are gaining popularity thanks to the entry level, easy to use, relatively cheap devices available, while serious professionals are indisputably relying on laser or structured light scanners for the best possible results. Considering only the Lidar scanners or even some structured light scanners that have a built in RGB camera as well, we can say that these devices can't produce data that has the necessary color detail for most heritage applications, since usually the built-in RGB camera is a low resolution sensor intended primarily to facilitate registering multiple scans into a complete 3D color model, and to give a generic colorization of the model. For cultural heritage applications, maybe except for visualization purposes in education, these models are not satisfactory. Thus, good quality, possibly color calibrated, high resolution RGB information has to be attached to the point cloud data. Lidar scanners that don't have an external camera attached will not capture RGB information in the same time with point cloud data, while structured light scanners that have a small sensor camera built in, will only capture poor quality color information. In both cases the solution is the same, RGB images have to be captured with a

separate device, even a full frame DSLR camera is quite commonly used for this task; and then fused with the point cloud.

Some recent works have shown that while the separate approaches may produce good partial results, the true potential is in combining multiple approaches. [73] used laser scanners and digital cameras for the documentation of desert palaces in the Jordan desert, while others also included CAD modeling in their work to complete the missing parts of the data [74]. An effective workflow using the combination of these three techniques was presented for 3D modeling of castles [75].

From a technical point of view the main challenge in fusing high resolution color calibrated RGB images with the 3D data is the estimation of the camera's relative pose to the reference 3D coordinate system. In the computer vision community many solutions are available solving this problem based on: finding point or line correspondences between the two domains [49], using mutual information [76], and large number of solutions relying on specific artificial landmarks or markers [53]. There are also expensive software solutions (*e.g.* [69] used Innov Metric Polyworks, [73] used Photomodeler) that solve this problem. However, these also require good quality RGB information in the 3D data, hence a pure geometric data with no RGB information is not enough to solve the fusion.

In contrast, our method works without color information in the 3D data and uses regions instead of matching key-points, which can be easier to detect in case of cultural heritage objects with homogeneous surface paintings. One region visible on both the 2D images and the 3D point cloud is already enough to solve the pose estimation, but with more regions the method becomes more robust [77]. In 2D, these regions can be easily segmented using standard segmentation methods, while in 3D, they can usually be segmented based on the 3D model's surface parameters or based on color information, if it is available. This means that we don't necessarily use color information stored with the 3D pointcloud, so an inexpensive device could also be used for data acquisition. The 2D images can also be acquired by any RGB camera, that can be calibrated using a free calibration Toolbox. Thus our workflow expects 2D color calibrated images, the camera's internal parameters and a 3D pointcloud with or without intensity information. In the application presented in Chapter 3.3 a refinement step is also proposed for the pipeline, that relies on available color information to further reduce the pose estimation error. Since the ICP based algorithm only makes use of the edge lines from the 3D RGB information, color accuracy and high resolution details are not needed, even a low resolution RGB information satisfies the needs of the refinement step, if the most prominent edge lines can be detected on it.

Since Lidar scanners are getting more often used for capturing large structures, complex buildings, in the second application presented in Chapter 3.4 we focused on the 2D-3D data fusion with Lidar scanners, since they can produce a widely usable, precise metric 3D model. Considering the relative poses of all the cameras to the 3D model are already obtained, another challenge arises when dealing with hundreds of such images, that have to be fused with one common 3D model. This involves different problems, such as choosing the best view for each part of the model, blending information from different sources with possibly different exposition, and generating a consistent, easy to handle output file format that is easily interpretable. For this problem we proposed a camera selection algorithm, that can deal with large numbers of images captured with different cameras, while relying on relevant parameters such as focal length, resolution, sharpness, viewing angle to choose the best view for every surface of the model. The algorithm ranks all the cameras that satisfy the visibility constraint for each 3D point, then chooses the best one according to some rules.

3.1.3 Contributions

In Chapter 3 we propose a straightforward absolute pose estimation method which overcomes the majority of the point based methods' limitations, *i.e.* by not using any artificial marker or intensity information from the depth data. Instead, our method makes use of a segmented planar region from the 2D and 3D visual data and handles the absolute pose estimation problem as a nonlinear registration task. More specifically, inspired by the 2D registration framework presented in [78], for the central camera model we construct an overdetermined set of equations containing the unknown camera pose. However, the equations are constructed in a different way here due to the different dimensionality of the lidar and camera coordinate frames as well as the different camera model used for omnidirectional cameras. By solving this system of equation in the least squares sense by a standard *Levenberg-Marquardt* algorithm, we obtain the required set of parameters representing the camera pose. Since segmentation is required anyway in many real-life image analysis tasks, such regions may be available or straightforward to detect. The main advantage of the proposed method is the use of regions instead of point correspondence and a generic problem formulation which allows to treat several types of central cameras in the same framework, including perspective and omnidirectional as well. The method has been quantitatively evaluated on a large synthetic dataset and it proved to be robust and efficient in real-life situations.

For cultural heritage focused applications in Chapter 3.3 we propose a 3D-2D region based fusion algorithm, that solves the pose estimation problem with segmented region pairs, even if no intensity information is available in the 3D data. If intensity information is also available the proposed algorithm makes use of it to refine the pose parameters in a 2D edge-lines based ICP refinement step. We show on synthetic benchmarks the performance of our method, including the robustness against segmentation errors that can occur in real world situations. We also validate the method on real data test cases which confirms that with good quality input data we can achieve high quality results, as well as moderate errors in the 3D model are well tolerated.

In Chapter 3.4 we propose a complete pipeline to fuse individual Lidar scans and 2D camera images into a complete high resolution color 3D model of large buildings. Commercial software provided by Lidar manufacturers are limited to the rigid setup of a laser scanner and a camera attached to it, for which they can produce correctly colorized models that are usable in many applications. Unfortunately, in cultural heritage applications usually a higher level of detail is necessary, especially for some parts of the scene of major importance. For this reason we have to separate the camera from the scanner and capture fine details from closer viewpoints using different telephoto lenses as well. Thus the proposed workflow contains a specific step used to select an optimal camera image for each 3D region that has the best view of that surface based on different criteria. This way we can project images of arbitrary cameras onto the 3D data in an efficient way, wide angle images providing a good general colorization for most parts while close up shots and telephoto images provide better resolution for selected parts. The efficiency and quality of the method has been demonstrated on two large case studies: the documentation of the Reformed churches of Klížska Nemá (Kolozsnéma) and Šamorín (Somorja) in Slovakia.

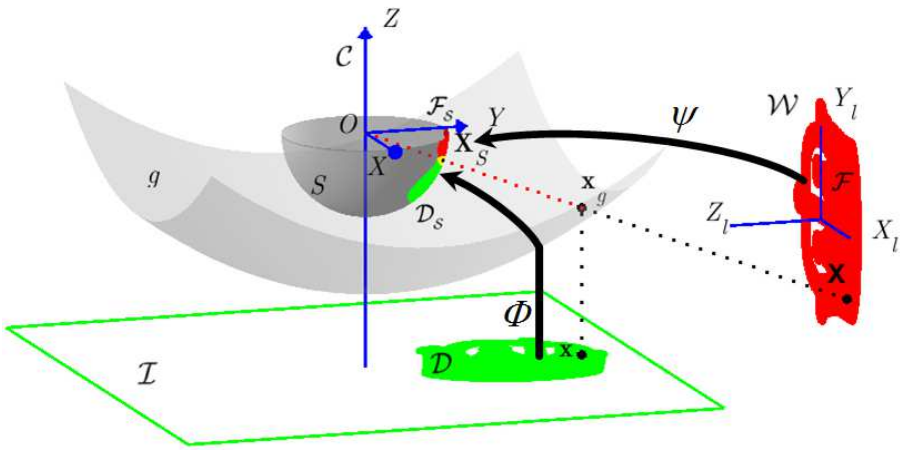

Figure 3.1. Spherical camera model and the projection of spherical patches $\mathcal{D}_S$ and $\mathcal{F}_S$.

3.2 Region-based Pose Estimation

Pose estimation consists in computing the position and orientation of a camera with respect to a 3D world coordinate system $\mathcal{W}$. Herein, we are interested in central cameras, where the projection rays intersect in a single point called projection center or single effective viewpoint. Typical examples include omnidirectional cameras as well as traditional perspective cameras. A broadly used unified model for central cameras represents a camera as a projection onto the surface of a unit sphere as described more detailed in Chapter 2.1.2 (see Fig. 2.3). The absolute pose of our central camera is defined as the rigid transformation $(\mathbf{R}, \mathbf{t}) : \mathcal{W} \rightarrow \mathcal{C}$ acting between the world coordinate frame $\mathcal{W}$ and the camera coordinate frame $\mathcal{C}$, that transforms points expressed in $\mathcal{W}$ into the coordinate system of the camera $\mathcal{C}$, while the internal projection function of the camera defines how 3D points are mapped from $\mathcal{C}$ onto the image plane $\mathcal{I}$.

Considering the generalized spherical camera model described in Chapter 2.1.2 we can clearly see that the projection of a 3D world point $\mathbf{X} = [X_1, X_2, X_3]^\top \in \mathbb{R}^3$ in the camera is basically a central projection onto $\mathcal{S}$ taking into account the extrinsic pose parameters ($\mathbf{R}$, $\mathbf{t}$). Thus for a world point $\mathbf{X}$ and its image $\mathbf{x} \in \mathcal{I}$, the following holds on the surface of $\mathcal{S}$:

$$\Phi(\mathbf{x}) = \mathbf{X}_\mathcal{S} = \Psi(\mathbf{X}) = \frac{\mathbf{R}\mathbf{X} + \mathbf{t}}{\|\mathbf{R}\mathbf{X} + \mathbf{t}\|} \tag{3.1}$$

A classical solution of the absolute pose problem is to establish a set of 2D-3D point matches using *e.g.* a special calibration target [59, 41], or feature-based correspondences and then solve for $(\mathbf{R}, \mathbf{t})$ via the minimization of some error function based on (3.1). However, in many practical applications, it is not possible to use a calibration target and most 3D data (*e.g.* point clouds recorded by a Lidar device) will only record depth information, which challenges feature-based point matching algorithms.

Therefore the question naturally arises: what can be done when neither a special target nor point correspondences are available? Herein, we present a solution for such challenging situations. In particular, we will show that by identifying a single planar region both in 3D and the camera image, the absolute pose can be calculated. Of course, this is just the necessary minimal configuration. More such regions are available, a more stable pose is obtained. Our solution is inspired by the 2D shape registration approach of Domokos

et al. [78], where the alignment of non-linear shape deformations are recovered via the solution of a special system of equations. Here, however, pose estimation yields a 2D-3D registration problem in case of a perspective camera and a restricted 3D-3D registration problem on the spherical surface for omnidirectional cameras. These cases thus require a different technique to construct the system of equations.

3.2.1 Absolute Pose of Spherical Cameras

For spherical cameras, we have to work on the surface of the unit sphere as it provides a representation independent of the camera internal parameters. Furthermore, since correspondences are not available, (3.1) cannot be used directly. However, individual point matches can be integrated out yielding the following integral equation [Tamas, Frohlich, Kato, 2014]:

$$\iint_{\mathcal{D}_\mathcal{S}} \mathbf{X}_\mathcal{S} \, \mathrm{d}\mathcal{D}_\mathcal{S} = \iint_{\mathcal{F}_\mathcal{S}} \mathbf{Z}_\mathcal{S} \, \mathrm{d}\mathcal{F}_\mathcal{S}, \tag{3.2}$$

where $\mathcal{D}_\mathcal{S}$ denotes the surface patch on $\mathcal{S}$ corresponding to the region $\mathcal{D}$ visible in the camera image $\mathcal{I}$, while $\mathcal{F}_\mathcal{S}$ is the surface patch of the corresponding 3D planar region $\mathcal{F}$ projected onto $\mathcal{S}$ by Ψ in (3.1) as shown in Fig. 3.1.

To get an explicit formula for the above surface integrals, the spherical patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ can be naturally parametrized via Φ and Ψ over the planar regions $\mathcal{D}$ and $\mathcal{F}$. Without loss of generality, we can assume that the third coordinate of $\mathbf{X} \in \mathcal{F}$ is 0, hence $\mathcal{D} \subset \mathbb{R}^2$, $\mathcal{F} \subset \mathbb{R}^2$; and $\forall \mathbf{X}_\mathcal{S} \in \mathcal{D}_\mathcal{S} : \mathbf{X}_\mathcal{S} = \Phi(\mathbf{x}), \mathbf{x} \in \mathcal{D}$ as well as $\forall \mathbf{Z}_\mathcal{S} \in \mathcal{F}_\mathcal{S} : \mathbf{Z}_\mathcal{S} = \Psi(\mathbf{X}), \mathbf{X} \in \mathcal{F}$ yielding the following form of (3.2) [Tamas, Frohlich, Kato, 2014]:

$$\iint_{\mathcal{D}} \Phi(\mathbf{x}) \left\| \frac{\partial \Phi}{\partial x_1} \times \frac{\partial \Phi}{\partial x_2} \right\| \mathrm{d}x_1 \, \mathrm{d}x_2 = \iint_{\mathcal{F}} \Psi(\mathbf{X}) \left\| \frac{\partial \Psi}{\partial X_1} \times \frac{\partial \Psi}{\partial X_2} \right\| \mathrm{d}X_1 \, \mathrm{d}X_2 \tag{3.3}$$

where the magnitude of the cross product of the partial derivatives is known as the surface element. The above equation corresponds to a system of 2 equations only, because a point on the surface $\mathcal{S}$ has 2 independent components. However, we have 6 pose parameters (3 rotation angles and 3 translation components). To construct more equations, we adopt the general mechanism from [78] and apply a function $\omega : \mathbb{R}^3 \to \mathbb{R}$ to both sides of the equation (3.1), yielding

$$\iint_{\mathcal{D}} \omega(\Phi(\mathbf{x})) \left\| \frac{\partial \Phi}{\partial x_1} \times \frac{\partial \Phi}{\partial x_2} \right\| \mathrm{d}x_1 \, \mathrm{d}x_2 = \iint_{\mathcal{F}} \omega(\Psi(\mathbf{X})) \left\| \frac{\partial \Psi}{\partial X_1} \times \frac{\partial \Psi}{\partial X_2} \right\| \mathrm{d}X_1 \, \mathrm{d}X_2 \tag{3.4}$$

Adopting a set of nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, each ω_i generates a new equation yielding a system of ℓ independent equations. Hence we are able to generate sufficiently many equations. The pose parameters $(\mathbf{R}, \mathbf{t})$ are then simply obtained as the solution of the nonlinear system of equations (3.4). In practice, an overdetermined system is constructed, which is then solved by minimizing the algebraic error in the *least squares sense* via a standard *Levenberg-Marquardt* algorithm. Although arbitrary ω_i functions could be used, power functions are computationally favorable [78, 77] as these can be computed in a recursive

manner:

$$\omega_i(\mathbf{X}_\mathcal{S}) = X_{S1}^{l_i} X_{S2}^{m_i} X_{S3}^{n_i}, \quad \text{with } 0 \leq l_i, m_i, n_i \leq 2 \text{ and } l_i + m_i + n_i \leq 3 \quad (3.5)$$

The summary of the proposed algorithm with the projection on the unit sphere is presented in Algorithm 1.

Algorithm 1 General form of the proposed pose estimation algorithm

Input: 3D point cloud and 2D binary image representing the same region, and the camera internal parameters

Output: External parameters of the camera as $\mathbf{R}$ and $\mathbf{t}$

1: Back-project the 2D image onto the unit sphere.
2: Back-project the 3D template onto the unit sphere.
3: Initialize the rotation matrix $\mathbf{R}$ from the centroids of the shapes on sphere.
4: Construct the system of equations of (3.2) with the polynomial ω_i functions.
5: Solve the set of nonlinear system of equation in (3.4) using the *Levenberg-Marquardt* algorithm

Note that the left hand side of (3.4) is constant, hence it has to be computed only once, but the right hand side has to be recomputed at each iteration of the least squares solver as it involves the unknown pose parameters, which is computationally rather expensive for larger regions. Therefore, in contrast to [Tamas, Frohlich, Kato, 2014] where the integrals on the 3D side in (3.4) were calculated over all points of the 3D region, let's consider a triangular mesh representation $\mathcal{F}^\triangle$ of the 3D planar region $\mathcal{F}$. Due to this representation, we only have to apply Ψ to the vertices $\{\mathbf{V}_i\}_{i=1}^V$ of the triangles in $\mathcal{F}^\triangle$, yielding a triangular representation [Frohlich, Tamas, Kato, 2019] of the spherical region $\mathcal{F}_\mathcal{S}^\triangle$ in terms of *spherical triangles*. The vertices $\{\mathbf{V}_{\mathcal{S},i}\}_{i=1}^V$ of $\mathcal{F}_\mathcal{S}^\triangle$ are obtained as

$$\forall i = 1, \ldots, V: \quad \mathbf{V}_{\mathcal{S},i} = \Psi(\mathbf{V}_i) \quad (3.6)$$

Due to this spherical mesh representation of $\mathcal{F}_\mathcal{S}$, we can rewrite the integral on the right hand side of (3.4) adopting ω_i from (3.5), yielding the following system of 17 equations [Frohlich, Tamas, Kato, 2019]:

$$\iint_\mathcal{D} \Phi_1^{l_i}(\mathbf{x}) \Phi_2^{m_i}(\mathbf{x}) \Phi_3^{n_i}(\mathbf{x}) \left\| \frac{\partial \Phi}{\partial x_1} \times \frac{\partial \Phi}{\partial x_2} \right\| \mathrm{d}x_1 \, \mathrm{d}x_2 \approx \sum_{\forall \triangle \in \mathcal{F}_\mathcal{S}^\triangle} \iint_\triangle Z_{\mathcal{S}1}^{l_i} Z_{\mathcal{S}2}^{m_i} Z_{\mathcal{S}3}^{n_i} \, \mathrm{d}\mathbf{Z}_\mathcal{S}, \quad (3.7)$$

where $\Phi = [\Phi_1, \Phi_2, \Phi_3]^\top$ denote the coordinate functions of $\Phi : \mathcal{I} \to \mathcal{S}$. Thus only the triangle vertices need to be projected onto $\mathcal{S}$, and the integral over these spherical triangles is calculated using the method presented in [79]. In our experiments, we used the Matlab implementation of John Burkardt[1].

The pose parameters are obtained by solving the system of equations (3.7) in the least squares sense. For an optimal estimate, it is important to ensure numerical normalization and a proper initialization. In contrast to [78], where this was achieved by normalizing the

[1] `https://people.sc.fsu.edu/~jburkardt/m_src/sphere_triangle_quad/`

input pixel coordinates into the unit square in the origin, in the above equation all point coordinates are on the unit sphere, hence data normalization is implicit. To guarantee an optimal least squares solution, initialization of the pose parameters is also important. In our case, a good initialization ensures that the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$, as shown in Fig. 3.1, overlap as much as possible. How to achieve this?

Initialization

The 3D data is given in the world coordinate frame $\mathcal{W}$, which may have an arbitrary orientation, that we have to roughly align with our camera. Thus the first step is to ensure that the camera is looking at the correct face of the surface in a correct orientation [Frohlich, Tamas, Kato, 2019]. This is achieved by applying a rotation $\mathbf{R}_0$ that aligns the normal of the 3D region $\mathcal{F}^\triangle$ with the Z axis, *i.e.* $\mathcal{F}^\triangle$ will be facing the camera, since according to the camera model $-Z$ is the optical axis. Then we also apply a translation $\mathbf{t}_0$ that brings the centroid of $\mathcal{F}^\triangle$ into $[0, 0, -1]^\top$, which puts the region into the $Z = -1$ plane. This is necessary to ensure that the plane doesn't intersect $\mathcal{S}$ while we initialize the pose parameters in the next step.

If there is a larger rotation around the Z axis, then the projected spherical patch $\mathcal{F}_\mathcal{S}^\triangle$ might be oriented very differently w.r.t. $\mathcal{D}_\mathcal{S}$. Using non-symmetric regions, this would not cause an issue for the iterative optimization to solve, but in other cases an additional apriori input might be needed, such as an approximate value for the vertical direction in the 3D coordinate system, which could be provided by different sensors, or might be specified for a dataset captured with a particular setup. Based on this extra information, we apply a rotation $\mathbf{R_z}$ around the Z axis that will roughly align the vertical direction to the camera's X axis, ensuring a correct vertical orientation of the projection.

To guarantee an optimal least squares solution, initialization of the pose parameters is also important [Frohlich, Tamas, Kato, 2019], which ensures that the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}^\triangle$ overlap as much as possible. This is achieved by computing the centroids of $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}^\triangle$, and initializing $\mathbf{R}$ as the rotation between them. Translation of the planar region $\mathcal{F}^\triangle$ along the direction of its normal vector will cause a scaling of $\mathcal{F}_\mathcal{S}^\triangle$ on the spherical surface. Hence an initial $\mathbf{t}$ is determined by translating $\mathcal{F}^\triangle$ along the axis going through the centroid of $\mathcal{F}_\mathcal{S}^\triangle$ such that the area of $\mathcal{F}_\mathcal{S}^\triangle$ becomes approximately equal to that of $\mathcal{D}_\mathcal{S}$.

Algorithm 2 Absolute pose estimation algorithm for spherical cameras

Input: The coefficients of g, 3D (triangulated) region $\mathcal{F}^\triangle$ and corresponding 2D region $\mathcal{D}$ as a binary image.

Output: The camera pose.

1: Produce the spherical patch $\mathcal{D}_\mathcal{S}$ from $\mathcal{D}$ using (2.6).
2: Produce $\mathcal{F}_\mathcal{S}^\triangle$ by prealigning $\mathcal{F}^\triangle$ as described in Chapter 3.2.1 using $(\mathbf{R_0}, \mathbf{t_0})$ and then $\mathbf{R_z}$, then back-projecting it onto the unit sphere $\mathcal{S}$ using (3.6).
3: Initialize $\mathbf{R}$ from the centroids of $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}^\triangle$ as in Chapter 3.2.1.
4: Initialize $\mathbf{t}$ by translating $\mathcal{F}^\triangle$ until the area of $\mathcal{F}_\mathcal{S}^\triangle$ and $\mathcal{D}_\mathcal{S}$ are approximately equal (see Chapter 3.2.1).
5: Construct the system of equations (3.7) and solve it for $(\mathbf{R}, \mathbf{t})$ using the *Levenberg-Marquardt* algorithm.
6: The absolute camera pose is then given as the composition of the transformations $(\mathbf{R_0}, \mathbf{t_0})$, $\mathbf{R_z}$, and $(\mathbf{R}, \mathbf{t})$.

The steps of the proposed algorithm [Frohlich, Tamas, Kato, 2019] for central spheri-

cal cameras using coplanar regions is summarized in Algorithm 2. For two or more non-coplanar regions, the algorithm starts similarly, by first using only one region pair for an initial pose estimation, as described in Algorithm 2. Then, starting from the obtained pose as an initial value, the system of equations is solved for all the available regions, which provides an overall optimal pose.

3.2.2 Absolute Pose of Perspective Cameras

A classical perspective camera sees the homogeneous world point $\mathbf{X}_{\mathcal{W}} = [X_1, X_2, X_3, 1]^\top$ as a homogeneous point $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, 1]^\top$ in the image plain obtained by a perspective projection $\mathbf{P}$:

$$\tilde{\mathbf{x}} \cong \mathbf{P}\mathbf{X}_{\mathcal{W}} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}_{\mathcal{W}}, \tag{3.8}$$

where $\mathbf{P}$ is the 3×4 camera matrix, which can be factored into the well known $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ form, where $\mathbf{K}$ is the 3×3 upper triangular *calibration* matrix containing the camera intrinsic parameters, while $[\mathbf{R}|\mathbf{t}]$ is the absolute pose acting between the world coordinate frame $\mathcal{W}$ and the camera frame $\mathcal{C}$.

As a central camera, the perspective camera can be represented by the spherical camera model presented in Chapter 2.1.2. Since we assume a calibrated camera, we can multiply both sides of (3.8) by $\mathbf{K}^{-1}$, yielding the normalized inhomogeneous image coordinates $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^2$:

$$\mathbf{x} \leftarrow \mathbf{K}^{-1}\tilde{\mathbf{x}} \cong \mathbf{K}^{-1}\mathbf{P}\mathbf{X}_{\mathcal{W}} = [\mathbf{R}|\mathbf{t}]\mathbf{X}_{\mathcal{W}}, \tag{3.9}$$

Denoting the normalized image by $\mathcal{I}$, the surface g in (2.4) will be $g \equiv \mathcal{I}$, hence the bijective mapping $\Phi : \mathcal{I} \to \mathcal{S}$ for a perspective camera becomes simply the unit vector of $\mathbf{x}$, as shown in Chapter 2.1.2:

$$\mathbf{X}_{\mathcal{S}} = \Phi(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{3.10}$$

Starting from the above spherical representation of our perspective camera, the whole method presented in the previous section applies without any change. However, it is computationally more favorable to work on the normalized image plane $\mathcal{I}$, because this way we can work with plain double integrals on $\mathcal{I}$ instead of surface integrals on $\mathcal{S}$. Hence applying a nonlinear function $\omega : \mathbb{R}^2 \to \mathbb{R}$ to both sides of (3.9) and integrating out individual point matches, we get [77]

$$\int_{\mathcal{D}} \omega(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{[\mathbf{R}|\mathbf{t}]\mathcal{F}} \omega(\mathbf{z})\,\mathrm{d}\mathbf{z}. \tag{3.11}$$

where $\mathcal{D}$ corresponds to the region visible in the normalized *camera* image $\mathcal{I}$ and $[\mathbf{R}|\mathbf{t}]\mathcal{F}$ is the image of the corresponding *3D planar region* projected by the normalized camera matrix $[\mathbf{R}|\mathbf{t}]$. Adopting a set of nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, each ω_i generates a new equation yielding a system of ℓ independent equations. Choosing power functions for ω_i [77]

$$\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i}, \quad 0 \le n_i, m_i \le 3 \text{ and } (n_i + m_i) \le 4, \tag{3.12}$$

and using a triangular mesh representation $\mathcal{F}^{\triangle}$ of the 3D region $\mathcal{F}$, we can adopt an efficient computational scheme. First, let us note that this particular choice of ω_i yields 13 equations, each containing the 2D geometric moments of the projected 3D region $[\mathbf{R}|\mathbf{t}]\mathcal{F}$. Therefore,

we can rewrite the integral over $[\mathbf{R}|\mathbf{t}]\mathcal{F}^{\triangle}$ adopting ω_i from (3.12) as [77]

$$\int_{\mathcal{D}} x_1^{n_i} x_2^{m_i} \,\mathrm{d}\mathbf{x} = \int_{[\mathbf{R}|\mathbf{t}]\mathcal{F}} z_1^{n_i} z_2^{m_i} \,\mathrm{d}\mathbf{z} \approx \sum_{\forall \triangle \in [\mathbf{R}|\mathbf{t}]\mathcal{F}^{\triangle}} \int_{\triangle} z_1^{n_i} z_2^{m_i} \,\mathrm{d}\mathbf{z}. \tag{3.13}$$

The latter approximation is due to the approximation of $\mathcal{F}$ by the discrete mesh $\mathcal{F}^{\triangle}$. The integrals over the triangles are various geometric moments which can be computed using efficient recursive formulas discussed hereafter.

2D Geometric Moments Calculation

Since many applications deal with 3D objects represented by a triangulated mesh surface, the efficient calculation of geometric moments is well researched for 3D [81, 80]. In the 2D case, however, most of the works concentrate on the geometric moments of simple digital planar shapes [84, 82, 83], and less work is addressing the case of triangulated 2D regions, with the possibility to calculate the geometric moments over the triangles of the region.

Since in our method we have a specific case, where a 3D triangulated region $\mathcal{F}^{\triangle}$ is projected onto the 2D image plane $\mathcal{I}$, where we need to calculate integrals over the regions $\mathcal{D} \subset \mathcal{I}$ and $[\mathbf{R}|\mathbf{t}]\mathcal{F}^{\triangle} \subset \mathcal{I}$, we can easily adopt the efficient recursive formulas proposed for geometric moments calculation over triangles in 3D and apply them to our 2D regions: Since our normalized image plane $\mathcal{I}$ is at $Z = 1$, the Z coordinate of the vertex points is a constant 1, hence the generic 3D formula for the (i, j, k) geometric moment of a surface S [80] becomes a plain 2D moment in our specific planar case [Frohlich, Tamas, Kato, 2019]:

$$M_{ijk} = \int_S x^i y^j z^k \,\mathrm{d}S = \int_S x^i y^j \,\mathrm{d}x \,\mathrm{d}y \tag{3.14}$$

as the last term of M_{ijk} will always be 1 regardless of the value of k. i and j are integers such that $i + j = N$ is the order of the moment.

Let us now see how to compute the integral on the right hand side of (3.13). The projected triangulated planar surface $[\mathbf{R}|\mathbf{t}]\mathcal{F}^{\triangle}$ consists of triangles T defined by vertices $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ that are oriented counterclockwise. The integral over this image region is simply the sum of the integrals over the triangles. Analytically, the integral over a triangle can be written as [85, 81]

$$\int_T z_1^i z_2^j \,\mathrm{d}\mathbf{z} = \frac{2\mathrm{area}(T) i! j!}{(i + j + 2)!} S_{ij}(T), \tag{3.15}$$

where

$$S_{ij}(T) = \sum_{(i_1+i_2+i_3=i)} \sum_{(j_1+j_2+j_3=j)} \Big(\frac{(i_1 + j_1)!}{i_1! j_1!} a_1^{i_1} a_2^{j_1} \frac{(i_2 + j_2)!}{i_2! j_2!} b_1^{i_2} b_2^{j_2} \frac{(i_3 + j_3)!}{i_3! j_3!} c_1^{i_3} c_2^{j_3} \Big). \tag{3.16}$$

Substituting (3.15) into (3.13), we get [Frohlich, Tamas, Kato, 2019]

$$\sum_{\forall T \in [\mathbf{R}|\mathbf{t}]\mathcal{F}^{\triangle}} \int_T z_1^i z_2^j \,\mathrm{d}\mathbf{z} = 2 \frac{i! j!}{(i + j + 2)!} \sum_T \mathrm{area}(T) S_{ij}(T) \tag{3.17}$$

where the signed area of triangle T is calculated as the magnitude of the cross product of

two edges:

$$\text{area}(T) = \frac{1}{2}\|(\mathbf{b}-\mathbf{a}) \times (\mathbf{c}-\mathbf{a})\| \tag{3.18}$$

As shown by [80] and then by [81], the computational complexity of the term $S_{ij}(T)$ can be greatly reduced from order $O(N^9)$ to order $O(N^3)$. Based on the final generating equations proposed by [81], we can write our generating equations for 2D domain as [Frohlich, Tamas, Kato, 2019]

$$S_{ij}(T) = \begin{cases} 0 & \text{if } i<0 \text{ or } j<0 \\ 1 & \text{if } i=j=0 \\ a_1 S_{i-1,j}(T) + a_2 S_{i,j-1}(T) & \\ \quad + D_{ij}(\mathbf{b},\mathbf{c}) & \text{otherwise} \end{cases} \tag{3.19}$$

with

$$D_{ij}(\mathbf{b},\mathbf{c}) = \begin{cases} 0 & \text{if } i<0 \text{ or } j<0 \\ 1 & \text{if } i=j=0 \\ b_1 D_{i-1,j}(\mathbf{b},\mathbf{c}) + b_2 D_{i,j-1}(\mathbf{b},\mathbf{c}) & \\ \quad + C_{ij}(\mathbf{c}) & \text{otherwise} \end{cases} \tag{3.20}$$

and

$$C_{ij}(\mathbf{c}) = \begin{cases} 0 & \text{if } i<0 \text{ or } j<0 \\ 1 & \text{if } i=j=0 \\ c_1 C_{i-1,j}(\mathbf{c}) + c_2 C_{i,j-1}(\mathbf{c}) & \text{otherwise} \end{cases} \tag{3.21}$$

Using only the equations (3.19)–(3.21), we can thus perform the exact computation of the contribution of every triangle to all the geometric moments of the image region in a very efficient way. The different quantities $C_{ij}(\mathbf{c})$, $D_{ij}(\mathbf{b},\mathbf{c})$, and $S_{ij}(T)$ are computed at order N from their values at order $N-1$ using the recursive formulas given above and they are initialized to 1 at order 0. The resulting $S_{ij}(T)$ are then multiplied by the area of the triangle T and summed up according to (3.17).

Initialization

As in Chapter 3.2.1, an initial rotation $\mathbf{R}_0$ is applied to ensure that the camera is looking at the correct face of the surface followed by an optional rotation $\mathbf{R}_z$ around the optical axis of the camera, that brings the up looking directional vector parallel to the camera's vertical axis, then apply a translation $\mathbf{t}_c$ to center the region in the origin. The initialization of the parameters $\mathbf{R}$ and $\mathbf{t}$ is done in a similar way as in Chapter 3.2.1: first the translation $\mathbf{t}$ along the Z axis is initialized such that the image region $\mathcal{D}$ and the projected 3D region are of the same size, then $\mathbf{R}$ is the rotation that brings the centroid of the projected 3D region close to the centroid of the corresponding image region $\mathcal{D}$ [Frohlich, Tamas, Kato, 2019].

The steps of the numerical implementation of the proposed method are presented in Algorithm 3. Note that for non-coplanar regions, as in Algorithm 2, we first use a single arbitrarily selected region for an initial pose estimation, then in a second step we solve the system using all the available regions, which provides an optimal pose estimate.

Algorithm 3 Absolute pose estimation algorithm for perspective cameras

Input: The calibration matrix $\mathbf{K}$, 3D triangulated region $\mathcal{F}^{\triangle}$ and corresponding 2D region $\mathcal{D}$ as a binary image.

Output: The camera pose.

1: Produce the normalized image $\mathcal{I}$ using $\mathbf{K^{-1}}$ as in (3.9)
2: Prealign the 3D region $\mathcal{F}^{\triangle}$ by rotating it first with $\mathbf{R_0}$ then with $\mathbf{R_z}$ as described in Chapter 3.2.2, then center the region in the origin using $\mathbf{t_c}$.
3: Initialize $\mathbf{t} = [0, 0, t_z]^\top$ such that the area of the regions are roughly the same (see Chapter 3.2.2).
4: Initialize $\mathbf{R}$ to ensure that the regions overlap in $\mathcal{I}$ as in Chapter 3.2.2.
5: Construct the system of equations (3.13) and solve it for $(\mathbf{R}, \mathbf{t})$ using the *Levenberg-Marquardt* algorithm.
6: The absolute camera pose is then given as the composition of the transformations $\mathbf{R}_0$, $\mathbf{R}_z$, $\mathbf{t}_c$, and $(\mathbf{R}, \mathbf{t})$.

3.2.3 Experimental Validation

Evaluation on Synthetic Data

For the quantitative evaluation of the proposed method, we generated different benchmark sets (of 1000 test cases each) using 25 template shapes as 3D planar regions and their images taken by virtual cameras. The 3D data is generated by placing $1/2/3$ 2D planar shapes with different orientation and distance in the 3D Euclidean space. Assuming that the longer side of a template shape is 1 m, we can express all translations in metric space. A set of 3D template scenes are obtained with $1/2/3$ planar regions that have a random relative distance of $\pm[1-2]$ m between each other and a random relative rotation of $\pm 30^\circ$.

Both in the perspective and omnidirectional case, a 2D image of the constructed 3D scenes was taken with a virtual camera using the internal parameters of a real 3Mpx 2376×1584 camera and a randomly generated absolute camera pose. The random rotation of the pose was in the range of $\pm 40^\circ$ around all three axes. The random translation was given in the range $\pm[0.5-2]$ m in horizontal and vertical directions and $2-6$ m in the optical axis direction for the perspective camera, while the omnidirectional camera was placed at half the distance, *i.e.* $1-3$ m in the direction of the optical axis, and $\pm[0.5-1]$ m in the X and Y axis directions to obtain approximately equal sized image regions for both type of cameras.

In practice, we cannot expect a perfect segmentation of the regions, neither in the 3D domain or on the 2D images, therefore the robustness against segmentation errors was also evaluated on synthetic data (see samples in Fig. 3.2): we randomly added or removed squares distributed uniformly around the boundary of the shapes, both in the 2D images and on the edges of the 3D planar regions, yielding different levels of segmentation error expressed as the percentage of the original shape's area. Using these images, we tested the robustness against 2D and 3D segmentation errors separately. For all these robustness tests, we show error plots for the maximum segmentation error levels where the median rotation error around any of the three axes was below 1°.

Theoretically, one single plane is sufficient to solve for the absolute pose, but it is clearly not robust enough. We have also found, that the robustness of the 1-plane minimal case is also influenced by the shape used: Symmetric or less compact shapes with smaller area and longer contour, and shapes with elongated thin parts often yield suboptimal results.

Figure 3.2. Examples of various amount of segmentation errors (*se*). First an omnidirectional image without *se*, then the same test with *se*$=12\%$, lastly the same template from a perspective test case with *se*$=20\%$.

However, such a solution can be used as an initialization for the solver with more regions. Adding one extra non-coplanar region increases the robustness by more than 4 times! We also remark, that the planarity of the regions is not strictly required. In fact, the equations remain true as long as the 3D surface has no self-occlusion from the camera viewpoint (see [Frohlich *et al.*, 2016] for a cultural heritage application). Of course, planarity guarantees that the equations remain true regardless of the viewpoint.

Since the proposed algorithms work with triangulated 3D data, the planar regions of the synthetic 3D scene were triangulated. For the perspective test cases a plain Delaunay triangulation of only the boundary points of the shapes were used, thus the mesh contains less but larger triangles, which are computationally favorable. For the spherical solver, however, a higher number of evenly sized triangles is desirable for a good surface approximation, which was produced by the `distmesh2D` function of [86] with the default parameters.

For a quantitative error measure, we used the rotation errors along the 3D coordinate axes and the difference between the ground truth $\mathbf{t}$ and estimated $\hat{\mathbf{t}}$ translation vectors as $\|\mathbf{t}-\hat{\mathbf{t}}\|$. Furthermore, as a region-based back-projection error, we also measured the percentage of non-overlapping area (denoted by δ) of the reference 3D shape back-projected onto the 2D image plane and of the 2D observation image. The algorithms were implemented in Matlab and all experiments were run on a standard six-core PC. A demo implementation is available online[2]. The average runtime of the algorithm varies from $1-3$ seconds in the perspective case to $4-7$ seconds in the omnidirectional case, without explicit code or input data optimization. Quantitative comparisons in terms of the various error plots are shown for each test case.

Omnidirectional Cameras

The results with 1, 2 and 3 non-coplanar regions using omnidirectional camera are presented in Fig. 3.3. In Fig. 3.3a - Fig. 3.3d, the rotation and translation errors for various test cases are presented. In the minimal case (*i.e.* 1 region), errors quickly increase, but using one more region stabilizes the solution: not only the error decreases but the number of correctly solved cases is also greatly increased. The δ error plot in Fig. 3.3e also confirms the robustness provided by more regions, while it has to be noted that with more regions the back-projection error does not improve in the way the pose parameter errors would imply, since even a smaller error in the pose yields larger non-overlapping area because of the longer boundary of the distinct regions.

[2] `http://www.inf.u-szeged.hu/~kato/software/`

While the perfect dataset is solved with median translation errors as low as 2 mm (see Fig. 3.3d), the error is increased by an order of magnitude, but still being under 3 cm, for regions corrupted with segmentation error. According to our previous experience [Tamas, Frohlich, Kato, 2014], a δ below 5% corresponds to a visually good result. Combining this metric with the rotation error limit of 1°, we conclude that our method is robust against segmentation errors of up to $\approx 12\%$ if at least 3 non-coplanar regions are used.

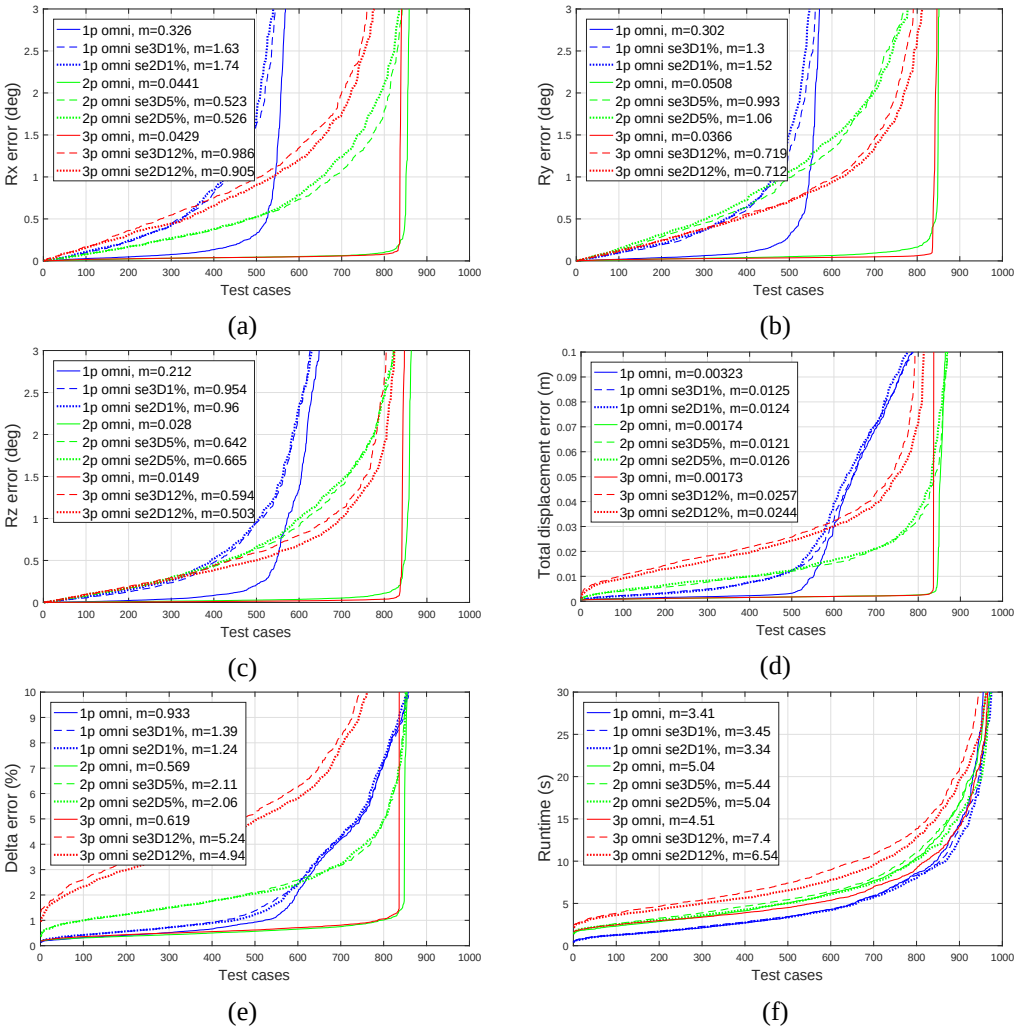

Figure 3.3. (a-f) Omnidirectional rotation errors along the X, Y, and Z axis, translation, δ error and runtime plots. m denotes median error, *se2D* and *se3D* stand for segmentation error on the 2D and 3D regions respectively (best viewed in color).

We have experimentally shown that the size of the spherical regions is greatly influencing the performance of the solver. While placing the camera closer to the scene produces larger spherical projections of the regions and the pose estimation becomes more robust, we aimed to use real world camera parameters instead, thus the camera-to-scene distance was limited. In our test cases, the median area of the spherical projections for the 1 and 3 region cases were 0.07 and 0.13 units respectively on the unit sphere.

For computing the spherical surface integrals, we compared two different approaches for the area approximation of the spherical regions. Our earlier approach is using standard numerical integration over the pixels projected onto the unit sphere [Tamas, Frohlich, Kato, 2014] as presented in Chapter 3.2.1 solving the system of equations in (3.4), while the more

recent approach in Algorithm 2 is integrating over spherical triangles instead as shown in (3.7). The δ error and runtime of these numerical schemes are compared in Fig. 3.4, which clearly shows that the CPU time of Algorithm 2 is an order of magnitude faster while the precision remains the same as for the earlier scheme in [Tamas, Frohlich, Kato, 2014].

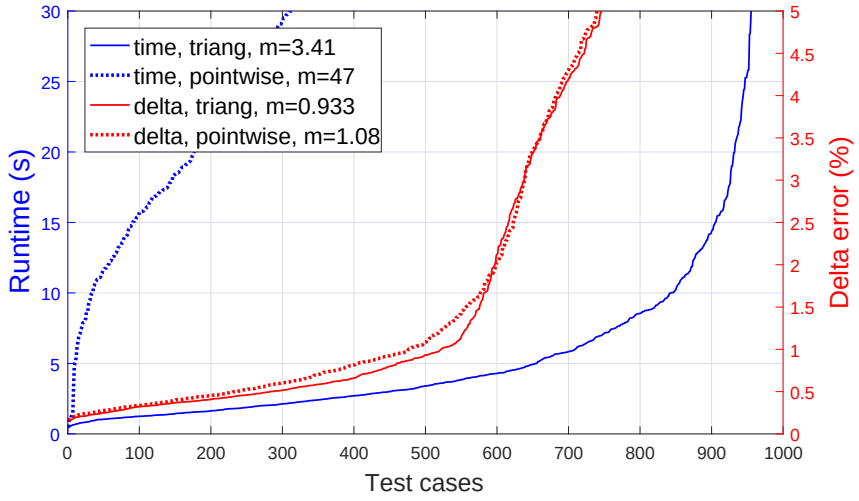

Figure 3.4. Backprojection (δ) errors and runtime comparison for point and triangle based spherical surface integral approximation on a 1 plane dataset (best viewed in color).

The algorithm's CPU runtime is shown in Fig. 3.3f, where the slightly increased runtime of the 3D segmentation error test cases (noted by *se3D*) is due to the triangulation of the corrupted planar regions, that increases the number of triangles around the edges and thus the computational time. Practically our algorithm can solve the pose estimation problem of an omnidirectional camera in ≈ 5 seconds using 2 regions.

Perspective Cameras

Pose estimation results using a perspective camera are presented in Fig. 3.5, including the same test cases with 1, 2 and 3 non-coplanar regions and with segmentation errors as in the omnidirectional case. The rotation and translation error plots in Fig. 3.5a - Fig. 3.5d clearly confirm the advantage of having more non-coplanar regions. The median translation error (see Fig. 3.5d) on the perfect dataset is as low as 2 mm, which increases by an order of magnitude in the presence of 20% segmentation error, but still being under 5 cm in case of 3 regions. The δ error plot in Fig. 3.5e also shows the robustness provided by the additional regions. Obviously, the back-projection error also increases in the presence of segmentation errors. However, as Fig. 3.5a - Fig. 3.5d shows, the actual pose parameters are considerably improved and the robustness greatly increases by using 1 or 2 extra non-coplanar regions.

The algorithm's CPU time on perspective test cases is shown in Fig. 3.5f. The increased runtime of the 3D segmentation error test cases (noted by *se3D*) is due to the triangulation of the corrupted planar regions, that greatly increases the total number of triangles and thus the computational time. Practically our algorithm can solve the pose estimation problem of a perspective camera in around 2.5 seconds using 2 regions.

As mentioned in Chapter 3.2.2, a perspective camera can also be represented by the spherical model developed in Chapter 3.2.3. However, as we have shown in the previous section, this model's main limitation is the small size of the spherical regions, because a

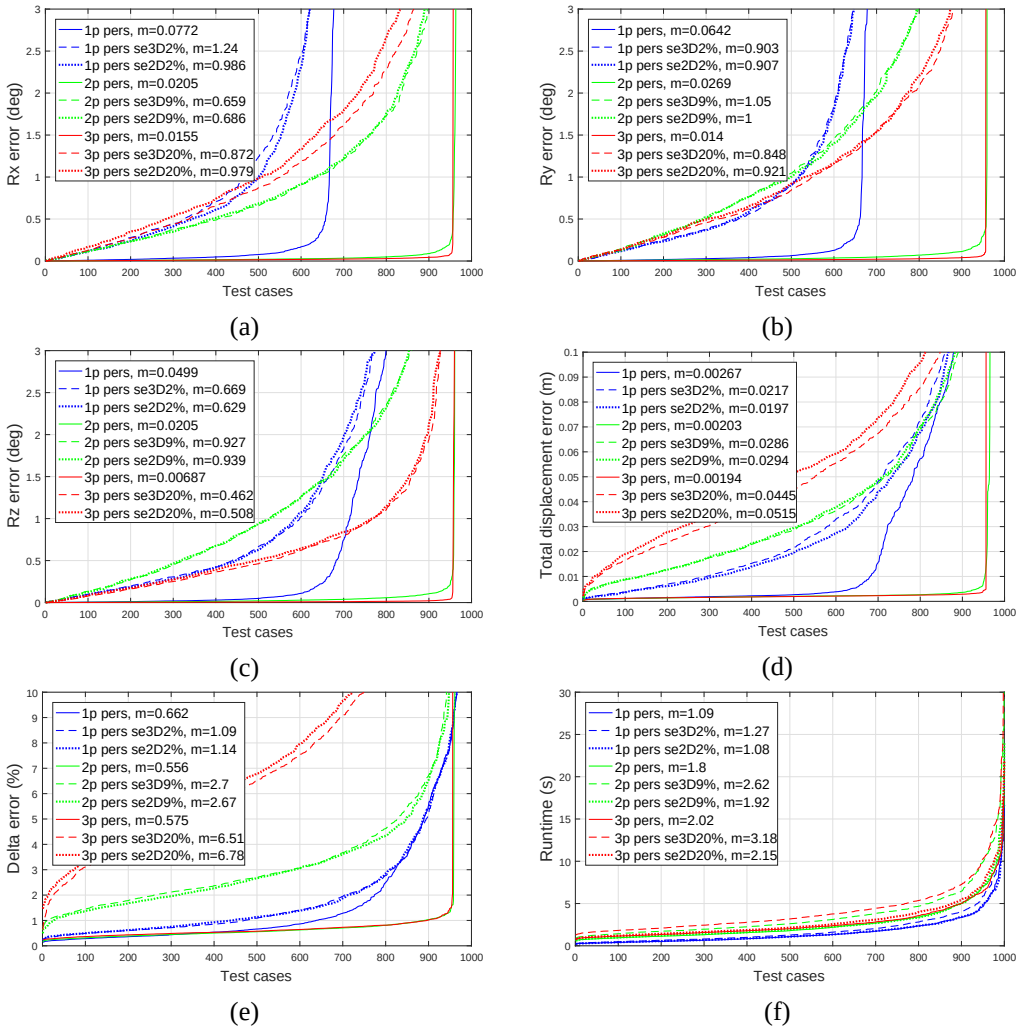

Figure 3.5. Perspective pose estimation results: rotation and translation errors, δ error and algorithm runtime plots. *se2D* stands for observation segmentation error, *se3D* for template side segmentation error and *m* for median values (best viewed in color).

perspective camera has a narrower field of view and has to be placed at a larger distance from the scene, to produce the same size of regions on the image. The resulting spherical projections of the planar regions in median are typically 4 times smaller than in the omnidirectional camera's case. Thus solving the perspective case using the spherical solver yields a degraded performance, as shown by the δ error plot in Fig. 3.6. Comparing the algorithm's runtime plot in Fig. 3.7 also shows that using the spherical solver for the perspective camera greatly increases the computing time due to the calculation of surface integrals on the sphere, which confirms the advantage of using the perspective solver proposed in Chapter 3.2.2, instead of a unified spherical solver.

To thoroughly evaluate our method on real world test cases, we used several different 3D data recorded by commercial as well as a custom built 3D laser range finder with corresponding 2D color images captured by commercial SLR and compact digital cameras with prior calibration and radial distortion removal. Whatever the source of the 2D-3D data is, the first step is the segmentation of planar region pairs used by our algorithm. There are several automated or semi-automated 2D segmentation algorithms in the literature including *e.g.* clustering, energy-based or region growing algorithms [87]. In this work, a simple

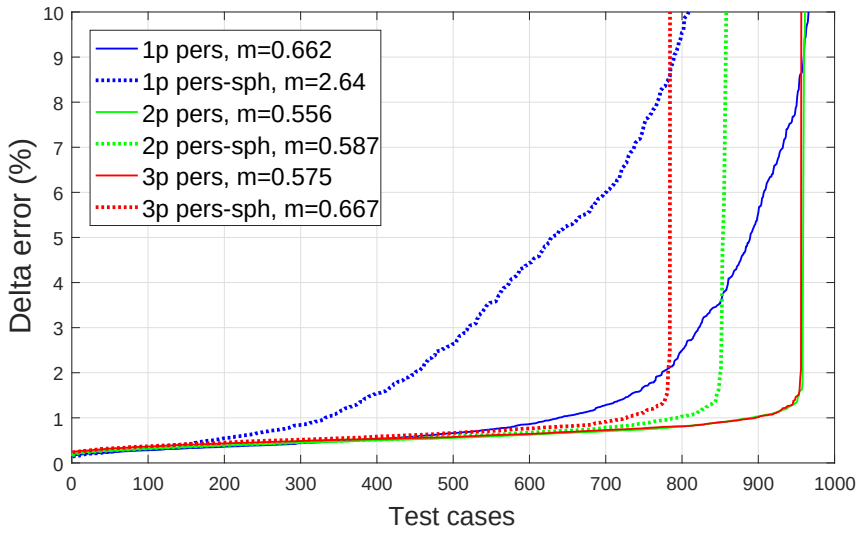

Figure 3.6. Perspective pose estimation δ errors comparing the normalized image plane and the spherical solutions. *m* stands for median values (best viewed in color).

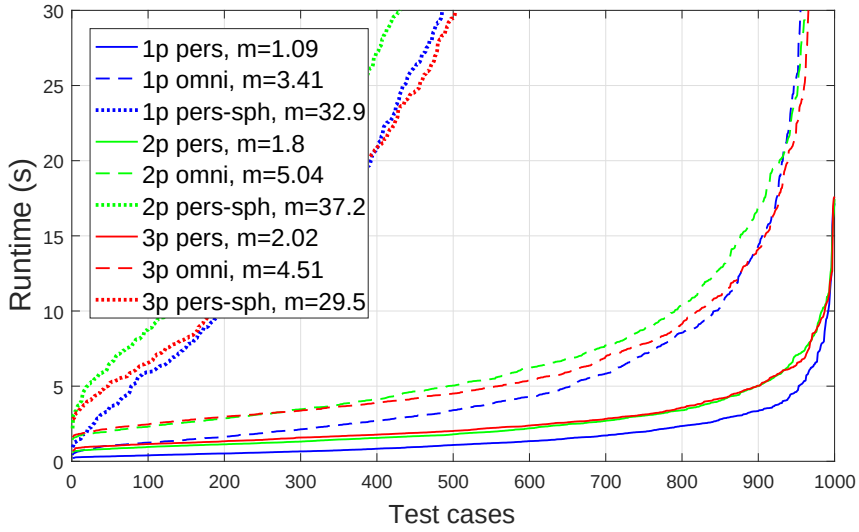

Figure 3.7. Runtime comparison on test cases without segmentation errors in the omnidirectional and perspective case, the latter using both the normalized image plane and the spherical solution. *m* stands for median values (best viewed in color).

region growing was used which proved to be robust enough in urban environment [88]. As for 3D segmentation, a number of point cloud segmentation methods are available, *e.g.* based on difference of normals [89] or robust segmentation [90]. Like in 2D, region growing based on surface normals gave stable results for extracting planar 3D regions in our experiments. Corresponding 2D-3D regions were simply selected during the seed selection of region growing as a one-click user input. We remark, however, that a fully automatic region correspondence could be implemented by detecting and extracting planar objects like windows [91] (see *e.g.* Fig. 3.8) which are typically planar surfaces present in urban scenes. If the segmented 3D region is a simple point cloud, the boundary of the region is detected using Alpha Shapes [92], which is then used for generating a triangular mesh (*i.e.* we do not rely on the Lidar resolution after segmentation). As in the synthetic case, for the omnidirectional case the method of [86] generated a uniform mesh, while for the perspective case a simple Delaunay triangulation was sufficient. The absolute pose obtained from Algorithm 2

or Algorithm 3 was used to fuse the depth and RGB data by projecting the images onto the 3D point cloud.

In Fig. 3.8, we show the fusion of an RGB perspective camera image and a sparse 3D point cloud recorded by a custom built 3D laser range finder containing a tilted Sick LMS200 ranger. The absolute pose of the camera was computed using Algorithm 3, which was then used to back-project the RGB image onto the 3D point cloud. Despite of the relatively large displacement between the camera and the Lidar, the absolute pose was successfully estimated.

Evaluation on Real Datasets

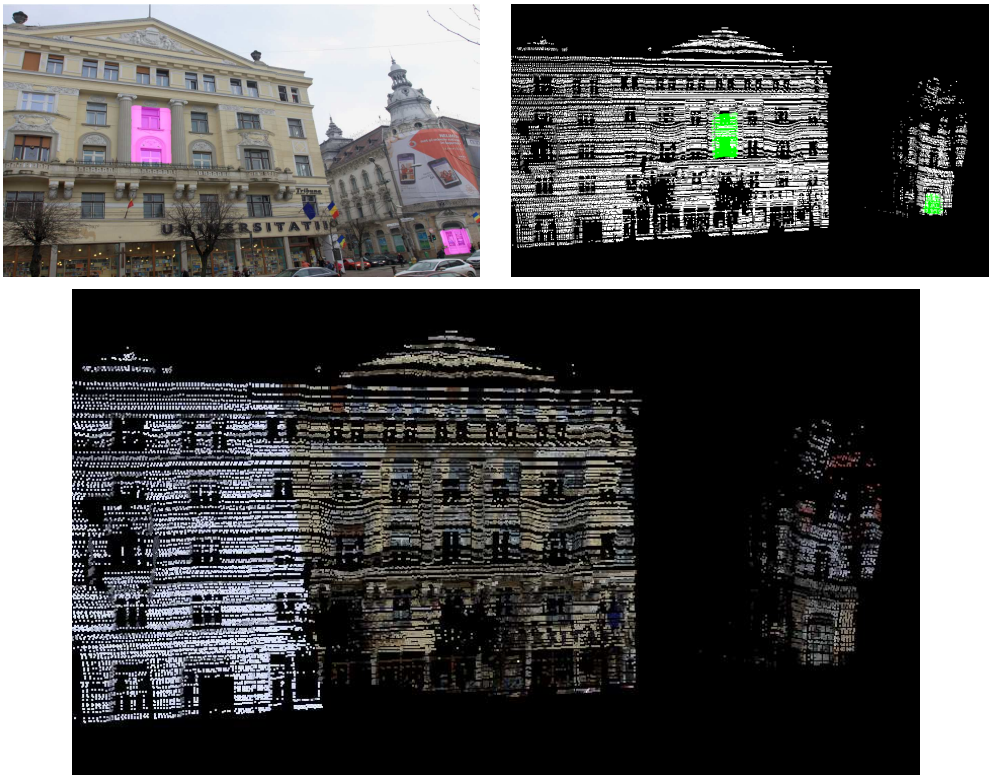

Figure 3.8. Pose estimation example with (left-right) central perspective camera and custom Lidar data: color 2D image (original frame) with corresponding regions (purple); 3D data with the segmented regions (green); color information overlaid on 3D data using the estimated camera pose (best viewed in color).

For the omnidirectional real data experiments we first tested the proposed method on 2D fisheye camera images and a 3D triangulated building model obtained by registering a set of sparse 3D laser scans recorded by a Velodyne HDL-64E mounted on a moving car [93] with a depth resolution up to 1 cm and an angular resolution up to $0.5°$. The best results were obtained by large non-coplanar regions. Such a test case is shown in Fig. 3.9, where the fish-eye camera image was reprojected onto the 3D surface using the absolute pose obtained by Algorithm 2. Note that in case of the omnidirectional cameras, even a relatively small rotation or translation error in the pose yields large differences in the non-linear distortions on the omnidirectional data. In spite of this sensitivity, Algorithm 2 proved to be robust enough as the segmented regions in Fig. 3.9 overlap well even if the total area of selected regions is relatively small compared to the whole image size.

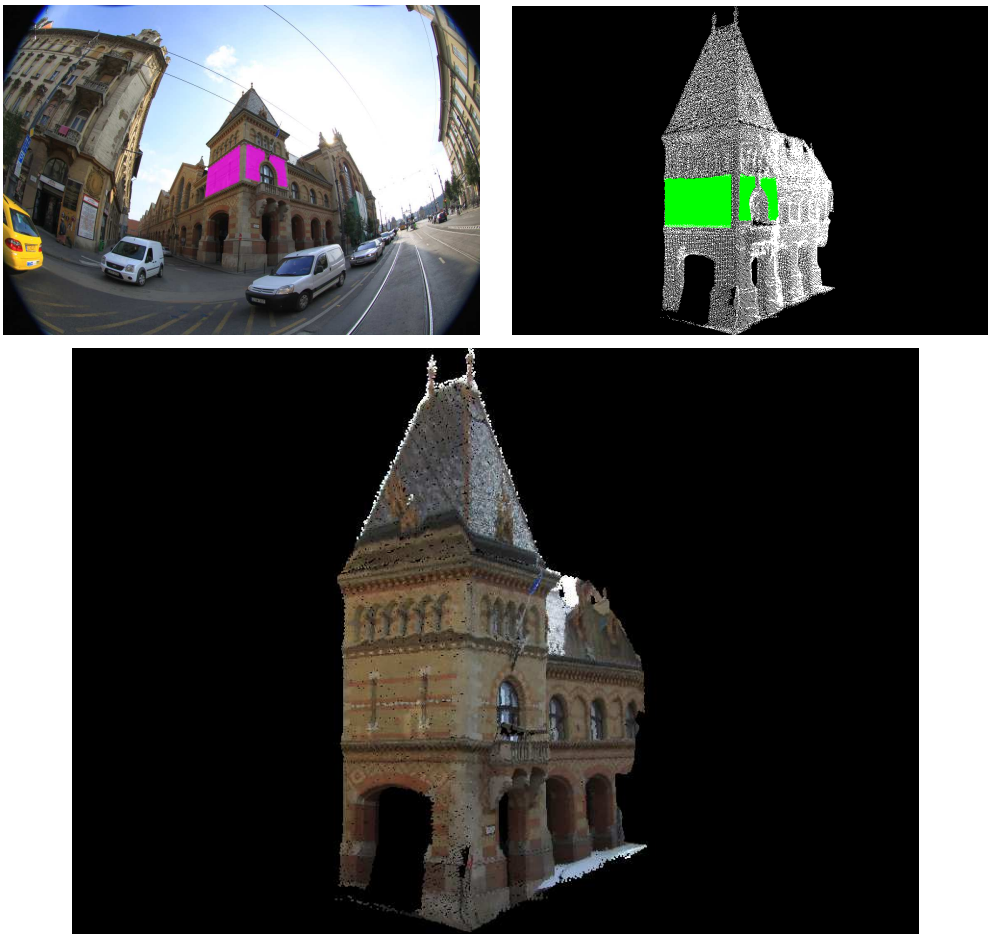

Figure 3.9. Pose estimation example with (left-right) central dioptric (fish-eye) and commercial (Velodyne) Lidar images: color 2D image (original frame) with corresponding regions (purple); 3D data with the segmented region (green); color information overlaid on 3D data using estimated pose parameters (best viewed in color).

Fusion result on a test case with a catadioptric-lidar camera pair is shown in Fig. 3.10. The omnidirectional image is captured by a commercial SLR camera with a catadioptric lens, while the 3D range data is provided by a custom built 3D laser range finder containing a tilted Sick LMS200 ranger, similar to the one described in [94] with an angular resolution up to half degree and a depth accuracy of 1 cm. The internal parameters of the omnidirectional camera were determined using the toolbox of [14]. The method proved to be robust against the segmentation errors caused by the low resolution of the image and also the noise in the 3D data, but a sufficiently large initial overlap between the regions was required for better results.

Finally, test cases with a high precision Riegl Lidar and different cameras are shown in Fig. 3.11 and Fig. 3.12. The static Riegl scanner has a range of 400 m with a depth precision of less than 0.5 cm and angular resolution up to 0.003°. In this dataset, the high density precise 3D model also includes the 3D positions of marker points that were set up on the building facade. Using these markers, we could evaluate the precision of our pose estimation by the forward projection of each marker from the 2D image into 3D space and then calculated the distance from their ground truth position.

For the omnidirectional case shown in Fig. 3.11, we used a full frame Canon EOS 5 DSLR camera with a 8 mm fish-eye lens. Segmenting only two simple, relatively small

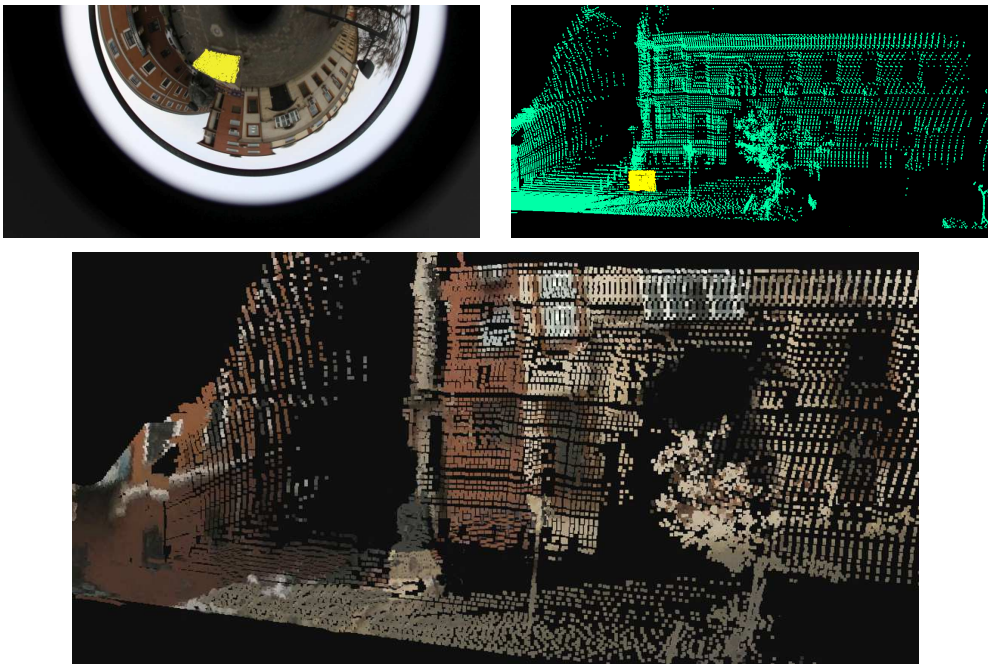

Figure 3.10. Catadioptric and lidar images with segmented area marked in yellow, and the fused images after pose estimation. (best viewed in color)

regions, the proposed Algorithm 2 estimated a precise pose with a forward projection mean error measured in the marker points of only 7 cm. The ground truth marker positions are visualized in green while the projected markers in red. Note that the camera-to-scene distance was $\approx$ 14 m in this case. For comparison, we also show in Table 3.1 the error of the absolute pose obtained by the state of the art UPnP [30] method, which directly used the ground truth marker positions as input 2D-3D point matches. In spite of working with perfect point correspondences, UPnP achieved only 2 cm better forward projection error in those marker points than our method which used inherently imperfect segmented region pairs.

For the perspective case in Fig. 3.12, we used a full frame Nikon DSLR camera with a wide field of view 20 mm lens, one of the typical RGB cameras that comes calibrated with these Riegl scanners. The mean forward projection error of the proposed Algorithm 3 measured in the marker points was 3 cm. The advantage of using multiple regions from differently oriented surfaces is clearly visible here. In Table 3.1, we compare our results to the factory calibration of the setup. It was interesting to find, that at 18 m distance from the wall, the factory calibration parameters produce 20 cm mean forward projection error, due to the interchangeable camera mounting system. Applying a marker based refinement to the calibration in the scanners own software, this can be reduced to 1.3 cm, which is only slightly better than our marker-less result achieved purely using 3 segmented region pairs.

The proposed Algorithm 3 was also tested with images taken by a flying DJI Phantom 3 drone. As can be seen in Fig. 3.12, the viewing angle of such a camera is very different from that of a ground level imaging device. Using two corresponding segmented regions was sufficient to estimate a correct pose with a mean forward projection error of 9cm, which is a good result considering the extreme angle of the camera and the camera-to-scene distance of $\approx$ 9 m. In comparison, the state of the art UPnP [30] and RPnP[29] methods using the high precision marker points as input 2D-3D point correspondences produced 2 cm and

Figure 3.11. Pose estimation example with omnidirectional camera image and dense Lidar data (left to right): color 2D image and 3D triangulated surface with corresponding segmented regions marked with purple and green respectively; lastly color information projected onto 3D data using the estimated extrinsic parameters, green dots mark the reference positions of the markers while red dots mark the projected positions (best viewed in color).

6 cm mean error, respectively.

The qualitative comparison of all the mentioned methods is presented in Table 3.1, where n/a stands for not available, since factory calibration parameters were only available in one case, and RPnP [29] cannot be used with omnidirectional cameras. Let us emphasize, that all the point-correspondence-based methods (except the Riegl factory parameters) rely on 2D-3D special markers, that were precisely measured in 3D and 2D. Thus to achieve these results with UPnP and RPnP, a careful setup of special markers in required before data acquisition, thus both 2D and 3D data capture must be performed at the same time. In contrast, the proposed method does not require any special target or setup, hence images recorded at different time can be fused as long as at least one planar region pair is available.

	UPnP	RPnP	Riegl	Riegl(fine)	Prop.
Omni	5	n/a	n/a	n/a	7
Pers. HR	0.9	4	20	1.3	3
Pers. Drone	2.2	6	n/a	n/a	9

Table 3.1. Comparisons on high resolution Lidar data in terms of the mean forward projection errors in marker points in cm. Note that results of UPnP [30], RPnP[29] and *Riegl(fine)* all rely on markers. *Riegl* stands for factory calibration, *Prop.* for the proposed method, and HR for high resolution full frame camera perspective test case.

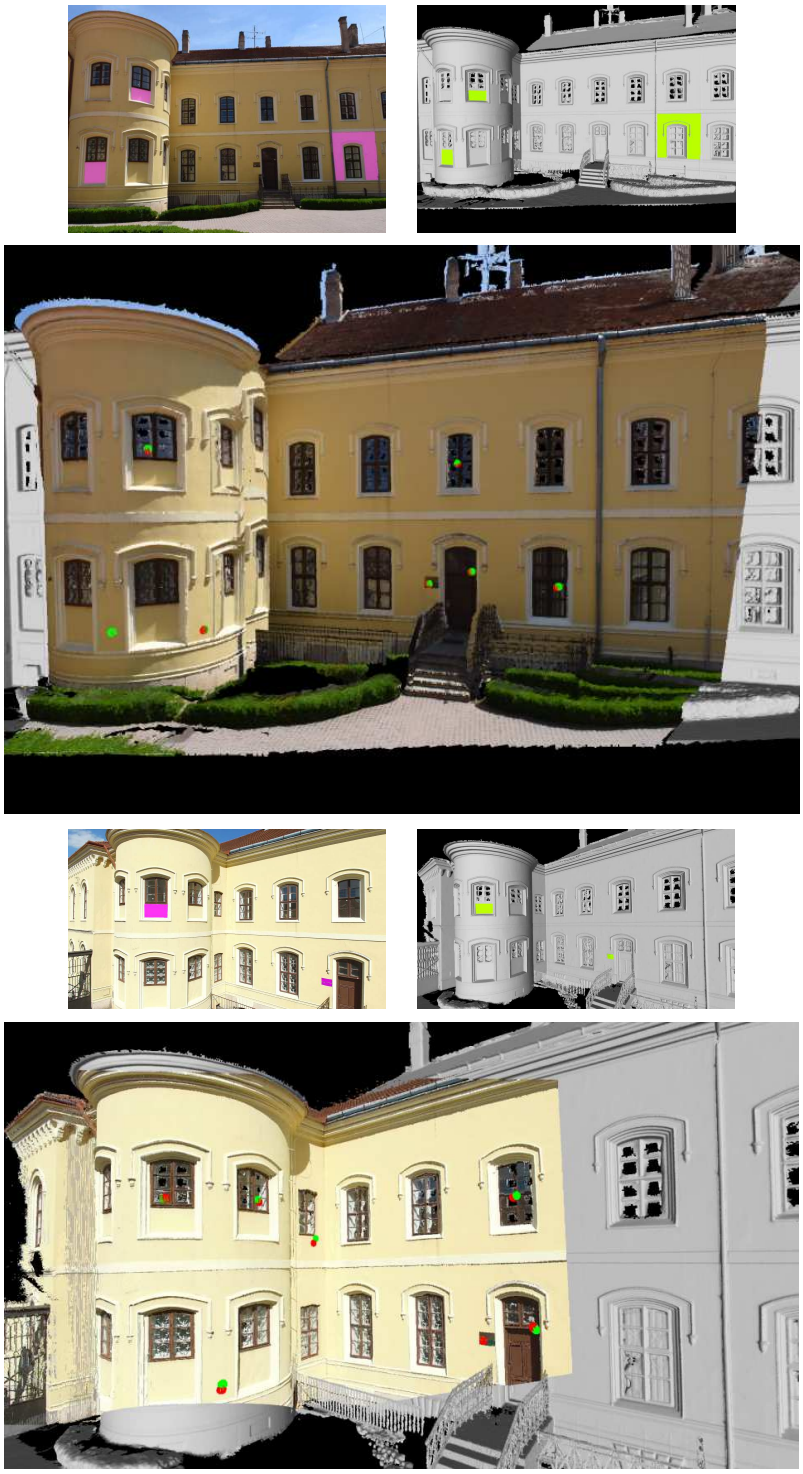

Figure 3.12. Pose estimation example with perspective cameras and dense Lidar data. First row: color 2D image and 3D triangulated surface with corresponding segmented regions marked with purple and green respectively. Second row: color information projected onto 3D data using the estimated pose, green dots mark the reference position of the markers while red dots mark the projected position. Top case: wide field of view camera; bottom case: normal field of view UAV camera. (best viewed in color).

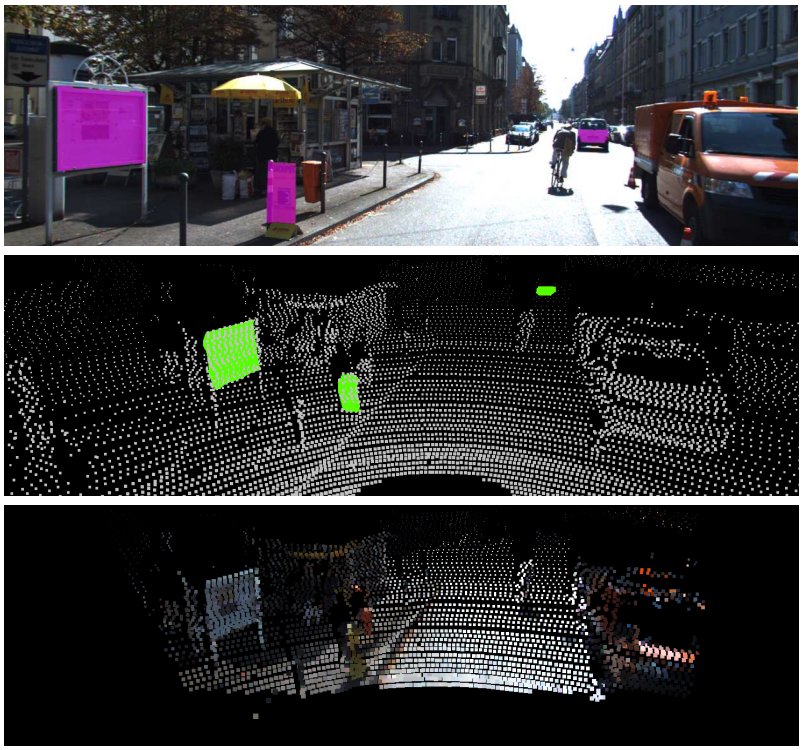

Figure 3.13. Pose estimation on the KITTI dataset, top: color 2D data with the selected regions (purple); middle: 3D data with the corresponding regions (green); bottom: color information overlaid on 3D data using the estimated camera pose.

	transl.	Rx	Ry	Rz	δ(%)	time(s)
Prop.	0.592	2.970	0.402	0.393	12.49	1.23
Norm.	0.441	0.522	4.740	0.745	74.01	166
Int.	0.397	3.254	4.826	1.543	46.77	147

Table 3.2. Comparative results with the proposed method (Prop), normal based MI(Norm)[45] and intensity based MI (Int)[45] in terms of translation(m), rotation(deg) and δ (for reference: δ for the ground truth pose is 9.49%) errors.

Algorithm Evaluation on the KITTI Dataset

Comparison with other camera pose estimation methods from the main literature could be performed only in a limited manner due to the fundamental differences of the proposed algorithm with respect to existing ones presented in Chapter 3.1. Methods using artificial markers like the ones described in [53, 57] were tested using the codes provided by the authors. The detailed comparisons are presented in our previous work [77]. Due to the limitations of [53, 57] on real datasets, we also tested the proposed method on the KITTI dataset [95] with available ground truth information. In Fig. 3.13 the extrinsic calibration of a color camera and sparse 3D Lidar data from the KITTI drive $nr = 5$ is shown. Using 3 segmented non-coplanar regions marked in purple and green in Fig. 3.13, the camera pose was estimated with the precision shown in Table 3.2.

For comparison, we used the mutual information based method described in [45] working on 3D data with intensity and normal information. The algorithm of [45] was run on the same 2D-3D data pair both in the normal based and intensity based configurations as pre-

sented in Fig. 3.13. The comparative results of absolute errors are also shown in Table 3.2. Note that while the algorithm of [45] is able to use multiple separate 2D-3D data pairs (if a sequence of such data is available with a rigid Lidar-camera setup like the KITTI dataset) to optimize the results, for a fair comparison we only provided the same single image frame and point cloud pair as the one that the proposed method was tested on. Since [45] is non-deterministic, the MI based results in Table 3.2 show the best ones out of 5 independent runs of the algorithm.

The results of the proposed method proved to be comparable to the results of [45], the normal based method being slightly better in the translation parameters, but worse in the rotation errors. Nevertheless the registration result of the proposed method visually was accurate, and the CPU implementation runtime was two orders of magnitude smaller than the GPU implementation of the mutual information method of [45].

3.3 2D-3D Fusion for Cultural Heritage Objects

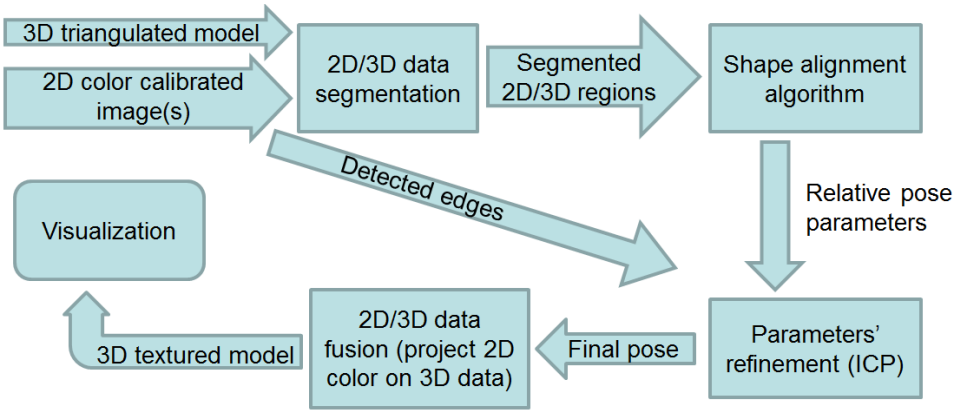

Figure 3.14. Workflow diagram of the proposed method.

In this chapter a workflow is proposed for 2D-3D data fusion, the diagram of which is shown in Fig. 3.14. The input consists of a 3D triangulated mesh and a set of 2D spectral images of the cultural heritage object. These images go through a 4-step processing pipeline [Frohlich *et al.*, 2016] in order to obtain a precise textured 3D model. We assume that the 2D cameras are color calibrated and their internal projection parameters are known, furthermore the acquired 3D point cloud has been preprocessed into a triangulated mesh (this is typically done by the 3D device's own software). In the following, we will present each processing step.

3.3.1 Segmentation (2D-3D)

As our method works with regions, we have to segment a corresponding set of regions both in the 2D images and 3D data. Since every test case is unique, we have to choose the segmentation method according to the surface properties. On the 2D images, any standard segmentation method (*e.g.* [96]) could be used (in our experiments, we used the *Fuzzy selection tool* of the free *Gimp* software). For the 3D data as well, we can choose based on the type of data that we have. When RGB information is available, we can simply use color based segmentation methods as in 2D. If it is not available, we can use 3D region growing, like the Minimum Covariance Determinant based algorithm [90] or interactive

graph cut like [97]. Manual selection can also be used, for example in our experiments the *Z-painting* tool of *Meshlab* has been used for interactive selection of regions, which works well regardless of the availability of RGB data. Let us emphasize, that no matter how many regions we extract from the data, they only have to correspond as a *whole*, a pairwise correspondence is not needed (even the number of regions can be different)! Hence, given a corresponding set of 2D regions $\{\mathcal{D}_i\}_{i=1}^{N}$ and 3D regions $\{\mathcal{F}_j\}_{j=1}^{M}$, they only have to satisfy the following constraint:

$$\mathcal{D} = \mathbf{P}\mathcal{F}, \text{ with } \mathcal{D} = \cup_{i=1}^{N}\mathcal{D}_i \text{ and } \mathcal{F} = \cup_{j=1}^{M}\mathcal{F}_i \tag{3.22}$$

where $\mathbf{P}$ is the camera projection matrix.

3.3.2 Pose Estimation

Given a corresponding set of segmented 2D-3D regions $\mathcal{D}$ and $\mathcal{F}$, we propose an extension of our plane-based Lidar-perspective camera pose estimation algorithm from Chapter 3.2 to the data fusion problem. While the method in Chapter 3.2 was used strictly on planar regions, we show that it can be extended to curved (but smooth) surfaces. This way it can be used in cultural heritage applications, since most of the objects, ceramics have non-planar but smooth regions.

Assuming that each of the segmented 3D regions $\{\mathcal{F}_j\}_{j=i}^{M}$ are smooth enough (*i.e.* they satisfy (3.22)), let us express a 3D point $\mathbf{X}_{\mathcal{W}}$ with its homogeneous world coordinates $\mathbf{X}_{\mathcal{W}} = (X_1, X_2, X_3, 1)^T$. The perspective camera sees the same world point $\mathbf{X}_{\mathcal{W}}$ as a homogeneous point $\mathbf{x} = (x_1, x_2, 1)^T$ in the image plain obtained by the perspective projection

$$\tilde{\mathbf{x}} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X} \tag{3.23}$$

As shown in Chapter 3.2.2, if we consider a calibrated camera, the effect of $\mathbf{K}$ can be inverted, resulting

$$\mathbf{x} = \mathbf{K}^{-1}\tilde{\mathbf{x}} = [\mathbf{R}|\mathbf{t}]\mathbf{X}_{\mathcal{W}} = \mathbf{P}\mathbf{X}_{\mathcal{W}} \tag{3.24}$$

thus the only unknown parameters are the 6 pose parameters (3 angles of rotation in $\mathbf{R}$, 3 components of the translation in $\mathbf{t}$). Classical solutions would establish a set of 2D-3D point matches (*e.g.* using special calibration targets or markers), and then solve for ($\mathbf{R}$,$\mathbf{t}$) via a system of equation based on (3.24).

However, in many cultural heritage applications, it is not always possible to attach markers to the object's delicate surface. Furthermore, the 3D scans and camera images might be acquired at different times, using different lighting conditions for optimal results. Our pose estimation method, based on the 2D shape registration approach presented in [77], proposes a solution in these challenging situations. Instead of using (3.24) directly, individual point matches are integrated out according to Chapter 3.2.2 yielding the following integral equation:

$$\int_{\mathcal{D}} \mathbf{x} d\mathbf{x} = \int_{\mathbf{P}\mathcal{F}} \mathbf{z} d\mathbf{z}, \tag{3.25}$$

where $\mathcal{D}$ corresponds to the regions visible in the *camera* image and $\mathbf{P}\mathcal{F}$ is the virtual image of the *3D region*s projected by $\mathbf{P}$. We can clearly see that the above integral equation stays valid for curved, smooth surfaces as well [Frohlich *et al.*, 2016], as long as $\mathcal{D}$ and $\mathcal{F}$ are satisfying (3.22) (*i.e.* no self-occlusion of points takes place). There are 2 issues with the above equation:

1. it corresponds to a system of 2 equations only, which is clearly not sufficient to solve for all 6 parameters of the camera pose;

2. the evaluation of the right hand side requires the explicit projection of the 3D regions $\mathcal{F}$, which might be computationally expensive.

To resolve 1), observe, that (3.24) remains valid when a function $\omega : \mathbb{R}^2 \to \mathbb{R}$ is acting on both sides of the equation [77]

$$\omega(\mathbf{x}) = \omega(\mathbf{P}\mathbf{X}_{\mathcal{W}}), \tag{3.26}$$

and the integral equation of (3.25) becomes

$$\int_{\mathcal{D}} \omega(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{P}\mathcal{F}} \omega(\mathbf{z}) d\mathbf{z}. \tag{3.27}$$

Thus adopting a set of nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, each ω_i generates a new equation yielding a system of ℓ independent equations. Hence we are able to generate sufficiently many equations. The ($\mathbf{R}$,$\mathbf{t}$) parameters of the camera pose are then simply obtained as the solution of the nonlinear system of equations (3.27). In practice, an overdetermined system is constructed, which is then solved by minimizing the algebraic error in the *least squares sense* via a standard *Levenberg-Marquardt* algorithm.

To resolve 2), let us choose power functions for ω_i

$$\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i}, \quad n_i \leq 3 \text{ and } m_i \leq 3, \tag{3.28}$$

which yields the 2D geometric moments of the projected 3D region $\mathbf{P}\mathcal{F}$, that can be computed efficiently. Since $\mathcal{F}$ consists of triangulated surface patches, their projection is a set $F^{\triangle}$ of triangulated planar patches, thus the final form of the equations becomes the same as in (3.13).

The integrals over the triangles are various geometric moments, which can be computed using the closed form formula presented in [77]

$$2 \sum_{k=0}^{p} \sum_{l=0}^{q} \frac{(-1)^{k+l} \binom{p}{k} \binom{q}{l} \nu_{kl}}{k+l+2} {z_1}_0^{p-k} {z_2}_0^{q-l} \tag{3.29}$$

where

$$\nu_{kl} = \sum_{i=0}^{k} \sum_{j=0}^{l} \frac{\binom{k}{i}\binom{l}{j}}{k-i+l-j+1} (z_{10} - z_{11})^i (z_{11} - z_{12})^{k-i} (z_{20} - z_{21})^j (z_{21} - z_{22})^{l-j} \tag{3.30}$$

with the notation z_{1i} and z_{2i}, $i = 0 \ldots 2$ being the vertices of the triangle.

Alternatively we can also use the recursive formulas presented in Chapter 3.2.2 that were adopted from the generic 3D formulas of [81] to 2D triangular regions.

3.3.3 ICP Refinement

In the previous step, we have obtained a camera pose by minimizing the *algebraic error* of the system in (3.13). Although this is already a good quality estimate, we can further refine it by minimizing a relevant *geometric error*. In the following, we will show how a standard Iterative Closest Point (ICP) [98] algorithm can be used, if color information,

even if it is of poor quality, is also available at each 3D point. In our workflow, ICP is used to align the 3D edge lines' projection with the 2D edge map (denoted by $\mathbf{x_e}$) of the camera image [Frohlich *et al.*, 2016]. There are different approaches to detect edges in a 3D pointcloud based on geometric properties, but for our purpose, we have to rely solely on the color information to be able to detect the same edges as in the 2D image. We have tackled this by simply projecting the 3D data onto an image with the initial camera pose using (3.23), then running Matlab's edge detection function on that image, resulting the edge points. The corresponding 3D points $\mathbf{X_e}$ will be the detected 3D edge points. The algorithm then iteratively projects the 3D $\mathbf{X_e}$ edge points using the current $\mathbf{K}[\mathbf{R}^n|\mathbf{t}^n]$ camera matrix, that has only the camera pose parameters $(\mathbf{R}^n, \mathbf{t}^n)$ changing between iterations, giving the reprojected edge points ${\mathbf{z_e}}^n$ at iteration n:

$$ {\mathbf{z_e}}^n = \mathbf{K}[\mathbf{R}^n|\mathbf{t}^n]\mathbf{X_e} \tag{3.31} $$

The ICP algorithm will align this ${\mathbf{z_e}}^n$ projection to $\mathbf{x_e}$, the edge map of the 2D image. We can clearly see that ICP will actually minimize the *backprojection error* this way.

3.3.4 Data Fusion

The final step of the workflow is the data fusion itself. Using the estimated relative pose and the calibration matrix of the camera, we can project (with (3.23)) the 3D points onto the 2D image. Since these do not necessarily project to exact pixel coordinates, we can interpolate the neighbouring pixels' color to find the best RGB value for every projected point. If we had multiple 2D input images, then we can fuse all images with the 3D data. For those 3D points, that are visible in more camera images, we have to decide which camera has the best view of it. For this purpose, let us calculate the normal vector $\mathbf{n}_i$ for each 3D point $\mathbf{X}_i$. In our experiments, we have used Meshlab's *Compute normals for point sets* function, which fits a local plane to every point's small neighborhood (10 neighbours). Then for every point $\mathbf{X}_i$ we compute the angle of its normal $\mathbf{n}_i$ with the orientation vector $\mathbf{c}_j$ of each camera's optical axis as

$$ \cos\theta = \frac{\mathbf{c_j} \cdot \mathbf{n_i}}{\|\mathbf{c_j}\|\|\mathbf{n_i}\|}, \tag{3.32} $$

and the camera image j with maximal $\cos\theta$ value is used to colorize the 3D point $\mathbf{X}_i$ [Frohlich *et al.*, 2016]. As a result, we get a good quality colored 3D model of the object. Since the 2D images are color calibrated, no color shift will appear, no transitions will be visible between regions that get RGB information from different images, if we assume a good uniform lighting was used when capturing the images. For easier examination of the results, we only used a single camera image for fusion in the test cases shown in Fig. 3.19.

3.3.5 Evaluation on Synthetic Data

For a quantitative and qualitative evaluation of the proposed pose estimation algorithm in Chapter 3.3.2, we have generated a benchmark set using 16 different shapes (such as in Fig. 3.15a). The 3D data was generated by projecting a 2D shape on a virtual spherical surface (having a Gaussian curvature of $K = 1/r^2 = 1/10000$). The 2D image of such an object was captured with a virtual camera having the intrinsic parameters of a standard 1Mpx camera and a random pose by rotating it with $[-25° - 25°]$ along all three axis and translating it randomly along all three axis with the maximum possible translation being

equal to the size of the object. A data set consists of 100 such images.

The results are presented in Fig. 3.15b - Fig. 3.15f. To evaluate the precision of the pose parameters, we backprojected the 3D points on the image plane, and calculated the percentage of the non-overlapping area (δ error) between the projection and the original observation. Ideally these should overlap perfectly, but experimentally we have found that 5% δ error or lower can be considered a correct result. We have also calculated the errors of the 3 rotation angles, and the translation error (see Fig. 3.15c) as the distance between reference and estimated position.

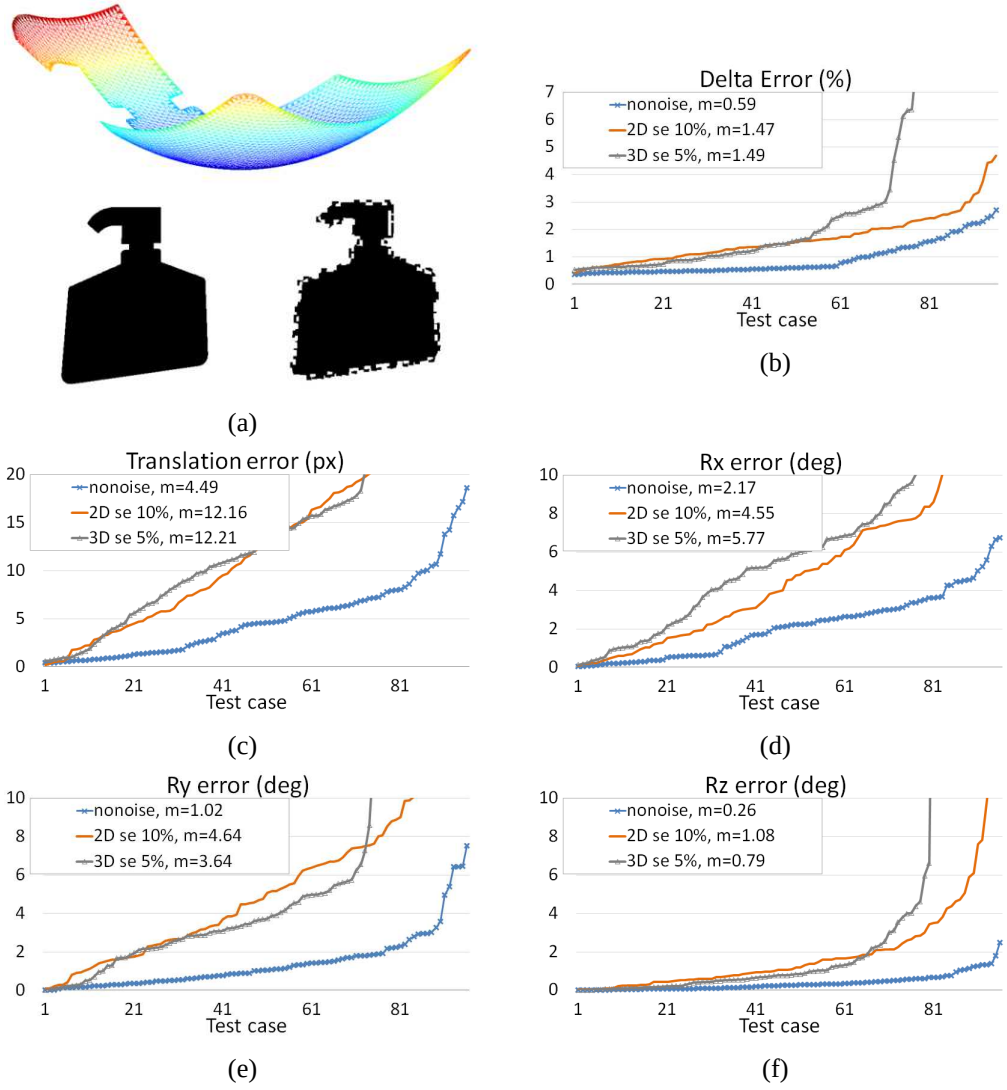

Figure 3.15. (a) Sample synthetic data, (b-f) error plots of the results on synthetic data (*2D se* stands for 2D segmentation error, *3D se* for segmentation error on the 3D data and *m* stands for median value).

Since in real cases both the 2D and 3D regions are affected by segmentation errors, we have also evaluated the robustness of our pose estimation method against such errors. For this purpose we have generated two different data sets: one with synthetically generated segmentation errors on 3D regions and another one with the 2D images being corrupted by it. An example synthetic data pair is shown in Fig. 3.15a, on top the 3D curved surface is shown, while below two images of the region, one with 10% simulated segmentation error. As we can see from the error plots in Fig. 3.15b - Fig. 3.15f, the method is more

robust for 2D segmentation errors. The same median δ error is achieved with 10% 2D segmentation error as with 5% error in the 3D segmentation, but with the 3D case we see more bad results. This also reflects on the rotation and translation error plots, while the median values are similar for the two cases, the number of incorrect results is higher in the 3D case. Nevertheless a median delta error of below 2% in using only a single curved region, is considered satisfactory.

3.3.6 Real Data Test Cases

We have verified our workflow on different real data test cases, of which 2 are presented in detail here: one using high precision data inputs, while the other using more affordable acquisition solutions.

The Chinese Warrior Test Case

The object used for this test case is a small (18 cm tall) figurine. The 2D images were taken with a calibrated Nikon D800 DSLR camera having a full frame 36 Mpx sensor, while the 3D data has been produced with a high precision marker based Structure-from-Motion software solution in strict laboratory conditions, giving us a perfect reference data in this case. While usual software solutions use markers or keypoints to produce such fused data, our method uses only the color images and a raw pointcloud (it doesn't even has to include RGB information!). In the first step of our workflow, we have to segment a few regions in 2D and 3D. Since the test object has a more complex, rugged surface, we have to concentrate on the smooth, well defined regions, where self-occlusion doesn't occur. Best choice in this case is segmenting the straps and bands on the clothes, since these are smooth regions, raised from their neighbors, with clearly visible ends. In 2D, a region growing segmentation tool was used, while in 3D an interactive selection method in Meshlab was adopted. Using the segmented data pairs, the second step estimates the pose of the camera relative to the 3D pointcloud with good precision. This is illustrated in Fig. 3.16 by backprojecting a few hand-picked 3D keypoints with the estimated camera pose to the 2D image (red dots) - which are close to their reference location (green dots). The measured average error was around $20 - 30$ px, which translates to approx. 1 mm real world error. In case we don't have access to intensity information in the 3D data, or if the object itself does not have a rich texture on its surface, then this is the final result. Note that if there is no intensity information, using a commercial software solution to align such data would also be challenging.

As in this test case we have color information too in the 3D input data, we can apply the ICP refinement step proposed in our workflow. The algorithm refines the relative pose based on the edge-map of the 2D image, and the projection of the 3D edge points. At this step, edges detected on smooth, mostly planar surfaces are desirable. To measure the benefits of using the ICP refinement step, we backprojected the same 3D keypoints and calculated the backprojection error. In Fig. 3.16, we can see that landmark points are projected closer to their correct location, reducing the average distance to 8px, equivalent to 0.2mm projection error. This can be considered good precision for most heritage applications. Final fused result from only a single camera image can be seen in Fig. 3.19.

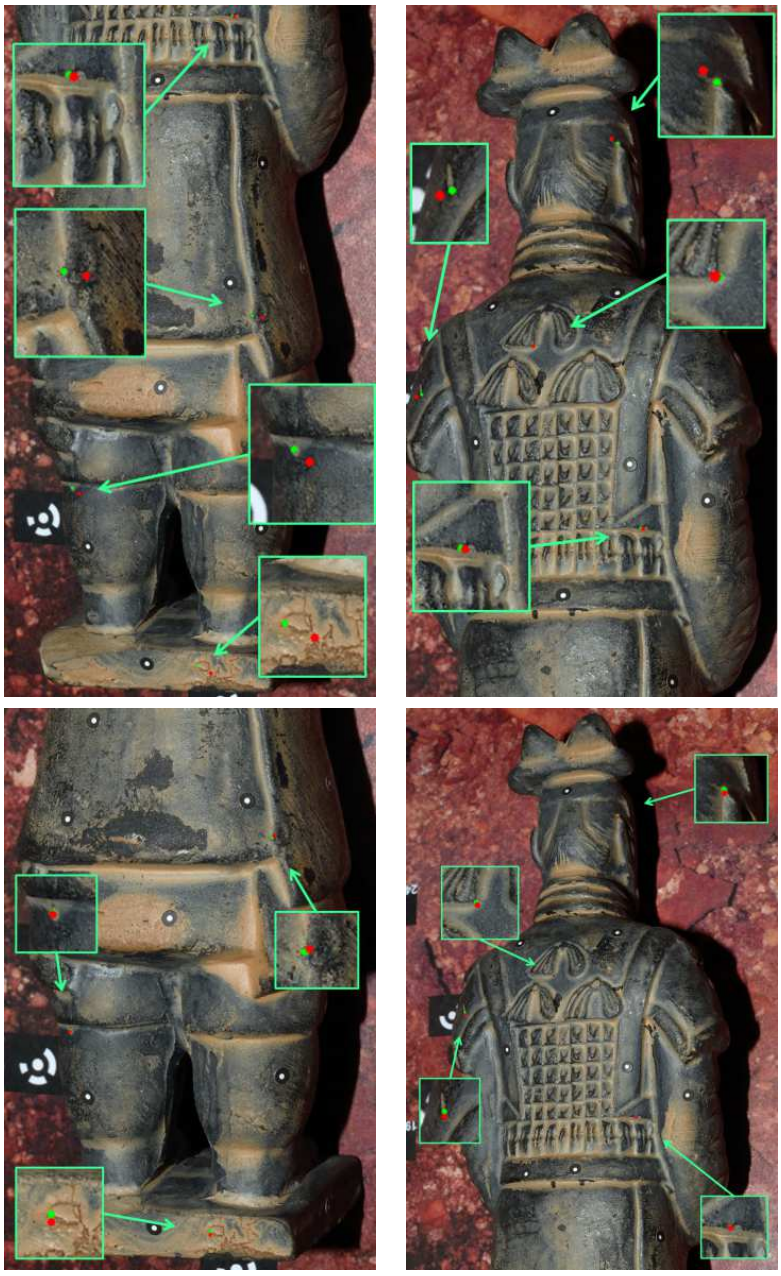

Figure 3.16. Precision of the region based pose estimation's results in first row, and results of ICP refinement in second row. Green dots are the reference locations while red dots are the back-projections of the 3D landmarks.

Ceramic Fragments Test Cases

The objects used in this test case are small fragments of ceramic bowls and vases. The 2D images were captured with a standard Canon 1000D DSLR camera, having 2.5Mpx resolution. The 3D data was produced by a handheld Artec Spider scanner and its bundled software. In this case, using a relatively cheap and easy to use scanner solution, we cannot expect perfect 3D data. The software uses a keypoint based algorithm to align partial scans and build the complete 3D model. Since the scanner only has a low resolution RGB camera built in, this process can get cumbersome in some situations. As we have found, even if the software produces a visually pleasing, watertight 3D model, it may lack precision. Of course a perfect alignment was not possible with these incorrect 3D data, but we have shown, that

in spite of the imperfect 3D model, our algorithm is robust enough to produce a good fused result. The segmented regions used for the pose estimation are shown in Fig. 3.17. The backprojection error of the two test cases can be seen in Fig. 3.18. The average error was 33px and 28px respectively.

Figure 3.17. 2D segmentation example.

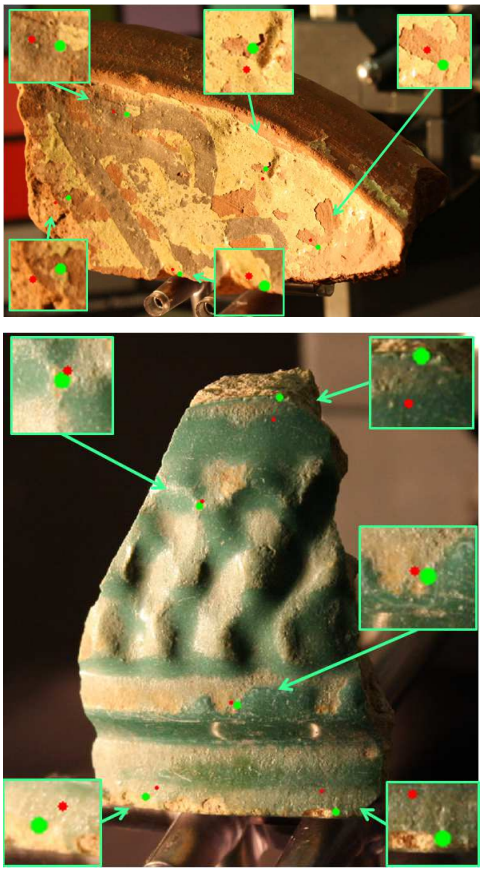

Figure 3.18. Final precision achieved using the ICP refinement step. Green dots are the selected specific landmarks, red dots are the back-projection of the same landmarks in 3D.

Figure 3.19. Final fusion results from single viewpoint, using the ICP refinement step. The ceramics 3D data are available from the authors affiliated to the Maison de l'Orient et de la Méditerranée.

3.4 Large Scale 2D-3D Fusion with Camera Selection

In this chapter we will describe the steps of the proposed processing workflow for large scale 2D-3D data fusion with optimal camera selection [Frohlich *et al.*, 2018]. The steps of the workflow are shown in Fig. 3.20 and detailed individually in the following.

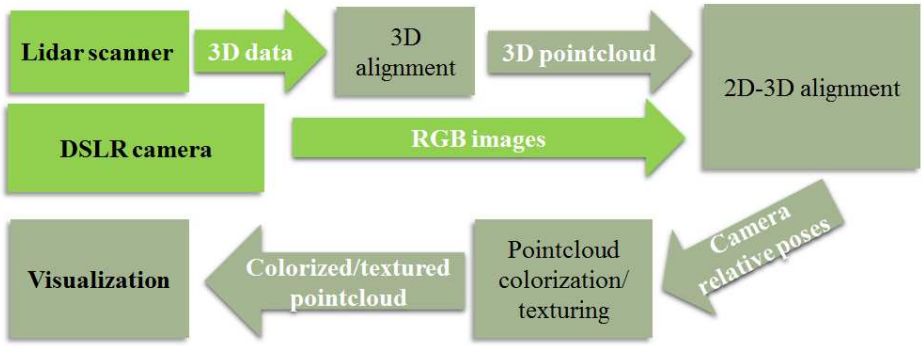

Figure 3.20. Workflow of our processing pipeline. Light green shows the input data.

3.4.1 Data Acquisition

For 3D measurements, we used the Riegl VZ-400 Lidar scanner with a horizontal angle resolution of 0.01° and a vertical angle resolution of 0.06°. A complete scan with 100° vertical and 360° horizontal field of view takes about 15 minutes, and produces a dense point cloud of $100 - 200$ million points, with a nominal depth precision of less than 5mm at distances below 400 m. Since a single scan only captures the surfaces visible from the Lidar's point of view, the whole surface of a complete building has to be scanned from multiple viewpoints. Usually, interior scenes are more complex than exterior ones, so these require a higher number of scans. For a more complex interior, a preliminary planning of scan positions is needed to provide the best coverage of the scene.

For 2D imaging, a Canon EOS 5D Mark III DSLR camera has been used with various optics. Actually, many Lidar manufacturers provide a solution to place a wide field of view camera on a rigid frame over the scanner, and let the scanner control the 2D capturing process as well. While this technique provides a reliable way to match 2D-3D data in subsequent processing steps, the common viewpoint constraint yields limited resolution of the 2D imagery for distant surfaces. In a typical cultural heritage application, the archeological site is far more complex, which cannot be captured in high detail from such a limited number of positions using a wide angle lens. Due to the fact that the Lidar scanner has a drastically higher range, being able to capture objects at up to 400 m with high resolution, it's not necessary to place the scanner closer to capture the small details, but with 2D imaging we can only produce high resolution images of small details (*e.g.* frescos) if we move the camera closer and use longer focal lengths for better reach. In addition, a 2D camera produces only sharp images of 3D objects located within its *depth of field* range. Hence for this reason it's mandatory to separate the camera from the scanner and take additional images from different viewpoints capturing all the fine details of the scene.

Thus the 3D-2D acquisition procedure typically consists of 2 stages: 1) acquisition of 3D Lidar scans together with a set of 2D images covering the complete 360° field of view from every scan position with a 24 mm wide lens; 2) acquisition of 2D images of all the important details from optimal viewpoints, using various focal lengths in the $70 -$

200 mm range. These high detail images would then be used to enhance the color and spatial resolution of the textured point cloud obtained from the wide angle images.

3.4.2 Point Cloud Alignment

The first step of 3D data processing is to register the Lidar scans into a common global coordinate frame. Let us consider a scanner that observes a 3D world point $\mathbf{X}_{\mathcal{W}}$ from different positions. In the first position S_1, the scanner will record the position $\mathbf{X_1}$ of the point $\mathbf{X}_{\mathcal{W}}$ in the Lidar's coordinate system that has its origin in the projection center of S_1. Moving the device to another position S_2 will measure position $\mathbf{X_2}$ for the same point $\mathbf{X}_{\mathcal{W}}$ in the coordinate system of S_2. The points $\mathbf{X_1}$ and $\mathbf{X_2}$ are related by a rotation $\mathbf{R}$ and translation $\mathbf{t}$:

$$\mathbf{X_1} = \mathbf{R}\mathbf{X_2} + \mathbf{t} \tag{3.33}$$

Given a sufficient number of $(\mathbf{X_1}, \mathbf{X_2})$ point pairs, one can easily compute the aligning rigid body transformation $(\mathbf{R}, \mathbf{t})$ between the scans S_1 and S_2. Actually the calculated transformation brings the coordinate system of S_2 into S_1. If we choose S_1 the global coordinate system, then we can align each S_i scan in the same way, bringing all the data into the same coordinate system, hence merging the partial scans into one single point cloud. For this task, we used the standard marker-based automatic registration algorithm available in the Lidar's software. As an alternative solution for outdoor scans, the software can also use the recorded GPS data instead of markers. If no markers nor GPS data is available, we can still do a registration by manually selecting sufficiently many corresponding $\mathbf{X_i}$ point pairs in the point clouds, but this will inherently be less precise.

3.4.3 Camera Pose Estimation

Next, we have to bring the 2D camera images into our world coordinate system established in the previous step.

The internal parameters of the camera can be easily estimated prior to data acquisition using standard camera calibration algorithms. Herein, we used the *Caltech Calibration Toolbox* [99]. However, the absolute pose (discussed in Chapter 2.2) has to be estimated for each image using *e.g.* standard feature-based methods relying on correspondences between a set of $\mathbf{X}_{\mathcal{W}\mathbf{i}}$ 3D points and $\mathbf{x_i}$ pixels [15]. In our experiments, we simply used the Lidar's software to compute camera poses from a given set of 2D-3D point pairs.

For a particular camera image, point correspondences can be obtained in a semi-automatic way using the markers detected in the point cloud and manually picking their corresponding pixel positions in the image. Images taken by wide angle lenses are likely to contain such markers detected in the Lidar scans, hence they can be reliably aligned this way. However, fine details are captured by telephoto lenses with a narrow field of view where markers may not be visible. Hence the selection of the corresponding points will become a manual task in both domains. We solved this by following a two-step procedure: first wide field of view images are processed in a semi-automatic way yielding a colorized point cloud. This data enables us then to manually select color based feature points in the tele images and the point cloud. As a result, we will have the pose for all images that capture different views at different resolution of the scene, giving us various possibilities for colorizing the point cloud.

3.4.4 Point Cloud Colorization

At this point, we have a complete metric 3D point cloud of the scene and a lot of different images taken from arbitrary viewpoints, but all registered with the global coordinate system. Since a particular 3D point $\mathbf{X}$ of the registered pointcloud may be seen by several cameras, the question naturally arises: How can we project the color information from the images onto the 3D points? In order to produce a high quality textured 3D model, several constraints have to be satisfied: the image used to colorize a point $\mathbf{X}$ should have a *sharp* image of $\mathbf{X}$ (*i.e.* it has to be within the camera's depth of field); the camera has to see the point $\mathbf{X}$ under an optimal angle (*i.e.* as close as possible to a perpendicular viewing angle) as well as the resolution around $\mathbf{X}$ should be as high as possible. Given a 3D point $\mathbf{X}$ and its projection in cameras $C_1 \dots C_n$ as pixels $\mathbf{x_1} \dots \mathbf{x_n}$, we can write the projection of the 3D point $\mathbf{X}$ in camera C_i using (3.23):

$$\mathbf{x_i} = \mathbf{K_i}[\mathbf{R_i}|\mathbf{t_i}]\mathbf{X} \tag{3.34}$$

where all the parameters are known by now. Thus the RGB color of the point $\mathbf{X}$ can be transferred from a particular camera image by making use of the above equation.

The commercial software solutions provided with Lidar devices usually assume, that the camera is used in a rigid setup with the scanner, having approximately the same viewpoint. In case of overlapping image regions the colors are simply averaged out for the corresponding 3D points. This approach is correct for this constrained setup, but becomes unusable when we separate the camera from the scanner, and place it in completely different positions, making the problem a more complex one. In this case every camera will have a completely different relative pose that has to be estimated, while in the standard commercial setup this is also reduced to only a change in the rotation $\mathbf{R}$ of the camera, which is directly recorded by the rotating Lidar. The visibility of the points from the camera viewpoint also has to be verified to avoid problems caused by occlusion. Using the commercial software in this special case, the more images are used the results can get more blurry because of the averaging of color information from cameras that had suboptimal view of a surface (*e.g.* camera at a bad angle, out of focus image region, camera too far away).

Therefore we propose a much more effective algorithm [Frohlich *et al.*, 2018] to tackle this problem, which will select for every 3D point one single camera that has the best view of it, *i.e.* it is not occluded, captured sharply, from the best angle and with the best resolution.

Visibility

First, we have to detect if a point $\mathbf{X}$ is visible from a camera or it is occluded. For this purpose, we have adopted the *Hidden Point Removal* operator [100]. It relies on the observation, that extracting the points that reside on the convex hull of a spherically flipped point cloud with respect to a given viewpoint, we get the visible points from that viewpoint. Let us consider the point cloud PC and the camera position C_1 from which PC is observed. Considering a sphere with the origin in C_1 and radius r constrained to include all the points of PC, spherical flipping will reflect all the points $\mathbf{X} \in PC$ with respect to the sphere by applying the following equation:

$$\hat{\mathbf{X}}_\mathbf{i} = \mathbf{X_i} + 2(r - \|\mathbf{X_i}\|)\frac{\mathbf{X_i}}{\|\mathbf{X_i}\|} \tag{3.35}$$

Visible points from C_1 are those that will reside on the convex hull of $\hat{PC} \cup C_1$ where $\hat{PC}$ denotes the transformed point cloud of PC [100]. Repeating this step for each camera C_i will give us the set of cameras from which a particular 3D point $\mathbf{X}$ is visible.

Sharpness

Next step is to verify if a point has a sharp image in the camera, only points that fall inside the *depth of field* of a camera C_i should be colorized from that camera image. The real world focus distance of the camera is not easily retrievable using only the image, but instead we can directly measure the upper and lower limits of the depth of field. Since for each image pixel we have the corresponding 3D point $\mathbf{X}$ and from the camera pose we can directly compute the camera-to-point distance, we only have to find the image regions that are in focus. For this purpose, we adopt the focus measure introduced by [101], which reflects the statistical properties of the wavelet transform coefficients in different high frequency subbands. Considering a 2D discrete wavelet transformation, in a single level transformation we will have four coefficient blocks, each $1/4$ of the size of the original image. The one noted with *LH* (Low High frequency) contains coefficients representing the vertical edges in the image, while the *HL* block shows horizontal edges, and the *HH* block containing high frequency components both in horizontal and vertical direction will represent the diagonal edges in the image. Using a randomly positioned window w over the original image, its corresponding operator windows in the single level wavelet transformation's *LH*, *HL*, and *HH* subbands are denoted by w_{LH}, w_{HL} and w_{HH} respectively, while the wavelet transform images in the subbands are denoted by W_{LH}, W_{HL} and W_{HH}. The focus measure operator is defined using the standard deviation of the wavelet coefficients as:

$$\begin{aligned} M^2_{WT} = \frac{1}{N_w}[& \sum_{(i,j)\in w_{LH}} (W_{LH}(i,j) - \mu_{LH})^2 + \\ & \sum_{(i,j)\in w_{HL}} (W_{HL}(i,j) - \mu_{HL})^2 + \\ & \sum_{(i,j)\in w_{HH}} (W_{HH}(i,j) - \mu_{HH})^2] \end{aligned} \tag{3.36}$$

where N_w is the number of pixels in w and μ is the expectation of the wavelet coefficients in each subband denoted with the corresponding subscript. We selected the windows w_s that had the focus measure M^2_{WT} above the experimentally determined threshold level of 1.1. We also experimentally determined an appropriate size for the window w, as a square window having $200px$ width, since on a full frame camera's 24Mpx image this is roughly similar in size to the focus detection squares that the camera uses, while smaller windows tend to often miss the sharp details on homogeneous regions.

Since the absolute pose of the camera C_i is known, we can simply calculate the average distance between the camera and the 3D points visible in window w_s as the average of the Euclidean distances from point to camera. Having a physical metric distance value $dist(w_s)$ assigned to each sharp window, let us create a histogram of the different distance values, and take the 5% and 95% percentiles of the distribution of the values to filter out possible

outliers. These values are an appropriate estimate for lowest and highest distance limits.

$$\begin{aligned} lowest\_dist &= hist(dist(w_s), @5\%) \\ highest\_dist &= hist(dist(w_s), @95\%) \end{aligned} \tag{3.37}$$

Points in this distance domain projected in camera C_i will have a sharp image. We apply these limits to filter out the cameras that don't see a given point sharply.

Viewing Angle

At this point, we have for each 3D point $\mathbf{X}$ a set of cameras assigned in which it's visible and in focus. As a next step we have to choose the one that sees the point from an optimal viewing angle and at highest resolution. Let us first calculate the angle between the surface normal $\mathbf{n_X}$ in $\mathbf{X}$ and the projection ray $\mathbf{o_{Xi}}$ pointing from $\mathbf{X}$ into the optical center of camera C_i. Since all the camera poses are known, the camera's projection center coordinates $\mathbf{c_i} = (x, y, z)^T$ are available. The surface normals in a point cloud can be calculated by different methods, like fitting local planes over a small neighborhood of the points [102], but these methods could have trouble detecting the correct orientation of the normals in case of large point clouds of complex scenes. Fortunately most Lidar scanners already provide the raw scan data with the correct normals in it, so we used this instead. The angle of these two vectors can be simply calculated using:

$$\theta = \arccos\left(\frac{\mathbf{n_X} \cdot \mathbf{o_{Xi}}}{\|\mathbf{n_X}\| \cdot \|\mathbf{o_{Xi}}\|}\right) \tag{3.38}$$

with $\mathbf{o_{Xi}} = \mathbf{X} - \mathbf{c_i}$ being the projection vector of point $\mathbf{X}$ into the i^{th} camera. The angles $|\theta| \in (0 \dots \pi/2)$ are the geometrically correct ones, as any other angle would mean that the camera is looking at the back side of the surface. Of course a mostly perpendicular view with small $|\theta|$ value is more favorable here.

Resolution

Next, we also check the projection resolution of the region, since a higher focal length camera can produce higher level of detail even from a larger distance, or a lower focal length camera from a closer position as well might have better resolution. We characterize the resolution of the projection of point $\mathbf{X_m}$ in the i^{th} camera as $res_{mi} = f_i/D_{mi}$, where f_i is the focal length of the camera and D_{mi} is the distance of camera i from point $\mathbf{X_m}$.

Selection

Then the final decision is taken by choosing the camera with the highest value of

$$dc_{mi} = res_{mi}/\theta' \tag{3.39}$$

where $\theta'$is the scaled version of angle θ into $\theta' \in [0 \dots 1)$ with 0 corresponding to the perpendicular view and 1 corresponding to the $\pi/2$ angle. dc_{mi} stands for the decision value of camera i with respect to the 3D point $\mathbf{X_m}$. The algorithmic overview of the method is summarized in Algorithm 4. Examples of the colorization with this vertex based color assignment can be seen in Fig. 3.22, Fig. 3.26 and Fig. 3.28.

3.4.5 Texture Mapping

The above presented algorithm provides a point cloud that has the best color assigned to each vertex point. In many applications it is desired to have good visual quality but with reduced data size, suitable for online streaming, storing or mobile applications. This can be achieved by using the triangular mesh that the scanner software provides us instead of just the point cloud. This also allows us to simplify the model by reducing the number of vertices defined, and visualizing surfaces instead. This also brings the benefit that we can map texture files to each triangle of the mesh, so instead of using the points' assigned color blended over the triangle face, what most software do when visualizing a colorized mesh, we can map a patch of the high resolution texture on it. This technique obviously will provide higher level of detail, and even on a reduced size data the apparent quality is almost the same. The size of the 3D data itself can be efficiently reduced by decimation algorithms [103], that will try to collapse multiple neighboring triangles on the same smooth surface into one single bigger triangle, reducing the necessary number of faces for smooth regions, while trying to keep a higher vertex number in the parts that are geometrically more complex.

Applying this to our proposed workflow we observe that for each point $\mathbf{X}$, we only need to store the corresponding texture coordinate in each camera image instead of the RGB value, so according to (3.34) we can extract the list of pixel coordinates $\mathbf{x_i}$ for each camera C_i. After that, going through the camera selection steps, when we already have a camera C^v assigned for each vertex v we can select for each face $F_j = (v_a, v_b, v_c)$ the best camera, simply by selecting the one that was commonly assigned to each vertex:

$$C^{F_j} = C^{v_a} \text{ iff } C^{v_a} = C^{v_b} = C^{v_c}$$

But what happens if the three vertices don't have a common camera assigned? This naive approach will cause issues on the edges of texture maps when it's the edge of the texture image and also at the boundary line between two textures. As an example for the latter, see Fig. 3.21, where on the left hand side we have a fresco that got textured from two different cameras, and at the boundary line there is a string of triangles (shown explicitly on the middle image) that didn't get either of the cameras assigned, since their vertices got assigned to different cameras.

Dealing with all these situations may be cumbersome, instead we adopted a new approach [Frohlich *et al.*, 2018] that iterates over all the triangles F of the mesh instead of the points. This way we are able to select different cameras for neighboring faces that have common vertices, and we are not limited to one single camera assigned per vertex point. The camera ranking steps presented in the previous section still remain valid and necessary, we only have to adapt the final step of the algorithm, in this case iterating over faces F of the mesh. For each face we look at the three C^{v_k} camera ranking lists assigned to each vertex, that contains the previously defined dc decision values for all C_i cameras:

$$C^{v_k} = dc_{ki}, \text{ where } k \in (a, b, c) \text{ and } i \in (1..n) \tag{3.40}$$

and select the camera C_i that got included in all three C^{v_k} lists and has the highest values

of dc. Assign this to face F_j:

$$C^{F_j} \in (C^{v_a} \cap C^{v_b} \cap C^{v_c}) \text{ where } dc = \max dc_{ki} \tag{3.41}$$

The single value assigned to c^{F_j} will be the camera index that was chosen for tha triangle. The corresponding texture coordinates are already available for all three vertices of the triangle, since we prepared them in the previous step. The data structure prepared this way can easily be written out in an ASCII Wavefront OBJ file based on its standard specifications [104]. The results obtained using the face based approach can be seen in Fig. 3.21 on the right hand side image, where different image textures are seamlessly connected. The algorithmic difference between the vertex based colorization and the face based texture mapping can be seen in Algorithm 4.

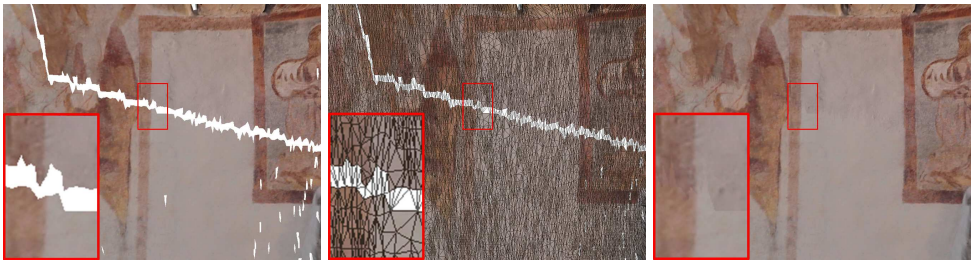

Figure 3.21. Issue with texture mapping on the boundary of different texture images (left and midle). Switching from vertex based to face based texture mapping the textures can join seamlessly (right).

Algorithm 4 The proposed camera selection algorithm

Input: A point cloud / triangular mesh and a set of images registered to it.
Output: A list with one camera assigned to every 3D point / face.

1: Considering a 3D point $\mathbf{X}$, first filter the list of cameras by visibility using (3.35).
2: Then filter the list of cameras by their depth of field (3.37) domain and the camera to point distance, keeping only those that have a sharp image of the point $\mathbf{X}$.
3: Using the remaining list of cameras, calculate for each the angle between the projection vector and the surface normal in point $\mathbf{X}$ using (3.38).
4: Also calculate each camera's projection resolution with respect to their distance from point $\mathbf{X}$ and the focal length.
5: Rank the cameras for each vertex v_k using (3.39), putting them in the list c^{v_k}.
6: Repeat steps 1-5 for all points of the point cloud.
7: **CASE** For vertex based colorization select the best camera from c^v for each vertex and assign the color seen by that camera according to (3.34).
CASE For texture mapping iterate over all the faces $F_j = (v_a, v_b, v_c)$, and select the camera that is best ranked in all three vertices' camera list c^{v_k} according to (3.41).

3.4.6 Experimental Results

The efficiency of the proposed method has been demonstrated on two large case studies. First, the documentation of the Reformed church of Somorja (Šamorín), then the documentation of the Reformed church in Kolozsnéma (Klížska Nemá), both of them located in Slovakia.

Reformed Church of Somorja

Somorja (Samaria, Sommerein, Zenthmaria) is a sacred edifice of great importance to the Upper Great Rye Island region. It also ranks among the most significant monuments of Christianity in the whole Carpathian Basin, and is standing proof of the high standard and prestige of medieval Hungarian Christian culture in Europe. A small chapel had stood on the site of the church sometime before the 11th century. The chapel was later continually expanded from the 11th through the 20th centuries. By the 14th century, the building's construction was supported by the likes of King Saint Stephen, King Béla III, Emperor Sigismund, King Matthias, King Wladyslaw II, and King Ludwig the Great.

The building had great sacred significance starting in the early Middle Ages, and contributed to a large degree the development of the municipality. Archaeological finds have revealed that, by 1521, the church had undergone twelve separate phases of reconstruction. The building's oldest part is the foundation of the Romanesque altar, situated below the current apse, which experienced continual additions since the 11th century. The tower has been standing in its current form since the 13th century. It is made entirely of brick, from its below-ground sections to the cap on the very top. The main nave's vaulted ceiling was built at the end of the 15th century in late Gothic style under King Wladyslaw II, in the style of the Prague castle.

The majority of the painted depictions of the main nave have not yet been revealed. Researchers have discovered that several ornate layers are still under the current plaster, the earliest of which dates back to the 11th century. During the Romanesque and Gothic era, i.e. in the 11th-12th and 14th centuries, the interior was completely covered with paintings, similar to Europe's other significant churches. On the northern wall of the apse, the earliest mural paintings appeared in a horizontal band depicting King Saint Stephen, King Béla III and Saint Adalbert the Bishop of Prague. To the right of these three portrayals, the painter depicted the most well-known scene from the life of Bishop Saint Martin when, as a Roman soldier, he dismounted his horse and handed half of his cloak to a shivering beggar. In a mural band under these images, the painter depicted the death of Mary. The imposing frescoes are an eloquent testimony to the significance of the town of Somorja in the 13th and 14th centuries. Few similar apse decorations have been preserved intact in Central and Western Europe. The depiction's intellectual message and iconographic statement praise a scholar theologian. According to the celestial vision of John the Apostle, these glorify the mysterious magnificence of the invisible God.

The pictures were covered by several layers of plaster for 600-700 years. What we see today is mostly the paintings'preparatory coating. The complete renovation of the exterior of the Reformed church in Somorja was supported by the Ministry of Human Resources of Hungary under Minister Zoltán Balog. Restoration work started in May 2014, and was completed in September 2015.

In Fig. 3.22 partial views are shown of the interior 3D model of the church in Šamorín. One of its invaluable heritages, the frescos on the sanctuary's ceiling, are visible in Fig. 3.22 on the first image. We used this fresco, depicting the coronation of Maria, to demonstrate the difference in resolution between an image taken with a telephoto lens, and a wide field of view image. In Fig. 3.23 we can see the region highlighted with red in Fig. 3.22 being cropped from a 70mm focal length (short telephoto lens) image and from a 24mm focal length camera image that captured a wide angle overview of the whole sanctuary. In Fig. 3.24 we can see the comparative results if we use these images for texture mapping on

Figure 3.22. Two views of the model colorized with the proposed vertex based method using a low number of images (24). The regions that were not sharply visible on any of the images are left white.

the triangular mesh. Regions that were not visible from any of the cameras are white.

Nevertheless, as we've shown previously, we need both types of images, to produce a highly detailed model, since we can only do a complete colorization of the model using wide field of view images, and when these already provide a color information for the points, we can register the high resolution images of the small details. As we can see in Fig. 3.24 this second registration step is also performed with good precision, since the two sides of the mesh are textured from the two images mentioned before, just for presentation purpose shown in a split way, and the transition between the regions is quite seamless. It is also noteworthy that just by being able to move the camera freely, we can get much higher resolution details even if using the same focal length lens by taking close up images. Capturing such images may be more intuitive for the non-professionals and they can still

Figure 3.23. Detail comparison of a wide and short telephoto image. On the left a crop of the 24mm camera image, on the right the same region as viewed by the 70mm camera.

Figure 3.24. A wide and short telephoto image used for texturing the mesh, viewed in a split way. The added detail of the tele image is clearly visible. White regions were not visible from any camera's point of view, therefore are not colorized.

provide an improvement to the 3D model's level of detail, as shown in Fig. 3.25, where the bottom of the pillar has been textured from a close up image, taken by the same camera that provided the wide angle images that textured the rest of the visible walls.

All the above mentioned comparison images show that the registration of the data is correct. Since our main interest was in the geometric registration and the camera selection process for the colorization, we didn't deal with color calibration in this work. Of course the correct color representation in such a cultural heritage documentation application is also a key factor, but standard solutions are available [105]. So instead we intentionally left all the images uncorrected in the color and lighting sense, this way the colorized model can give us a clue about which regions were colorized from different cameras, since these transitions are not blended in any way. It is well visible in Fig. 3.22 on the walls and on the floor that these kind of visible errors are only caused by the constantly changing illumination of the scene, the shadows, and the inhomogeneous lighting in some regions. If we examine closely the transitions between the different colorizations on the ground, we can see the correct alignment of the cameras, since the edge lines of the bricks are well matched. In Fig. 3.26 we can observe what improvements can be achieved by attempting to correct the white balance and exposure of the images in post processing, done by an unexperienced user. While the color tones are more similar between the different images, the most visible issue caused by the constantly changing lights and shadows would only be avoidable by using a controlled lighting setup during the acquisition.

One interesting use of a 3D model produced this way, is the possibility to illustrate historical stages of the buildings, for example by removing completely the organ from the

Figure 3.25. Model textured with distant images, while a close up image brings significant improvement in resolution, as seen on the bottom part of the pillar. Green line marks the texture boundary.

sanctuary (visible in Fig. 3.22), that was only added recently, we could visualize how the sanctuary could have looked like centuries ago. This kind of depiction is only possible on such 3D data. An example is shown in Fig. 3.27, it is noticeable how the windows, the walls and the paintings behind the organ get visible on the second image.

Reformed Church of Kolozsnéma

There are two separate theories related to the foundation of the reformed church located in the village of Kolozsnéma (Klížska Nemá). According to those theories the church could have been a Turkish mosque or a Catholic chapel, however most likely in the reality the church is a tower of a castle owned by the family Kolosfi, built approximately in 1375 at the age of Ludwig the Great. It can be assumed that after the devastation of the castle the church was built on its place and the crypt of Kolosfi's can be still found under the building keeping the possibility to perform an archaeological excavation in that area. A dream of the people, living in Kolozsnéma, about the magnification of the church came true during the ministry of Ferenc Borza (1784 to 1794). The congregation has renovated their church in 1819 and during the construction work a small window has been structured at the western part of the building in order to make the indoor part brighter. Due to the fact that the place dedicated for the men was not big enough they have built a gallery at the western part and in the same time 2 brand new windows have been constructed to keep the necessary level of the natural brightness inside. Unfortunately a huge fire has destroyed all of the buildings belonging to the congregation at the time of Albert Kőváry who was the last pastor of the village living on site. In the fire almost all of the assets have been damaged like the church, the school, the bowles as well as both of the bells. The congregation was depressed by the calamity, however they did not give up. The damaged church have been corrected on the 23rd of May 1858 and later in 1928 and 1929 the church has been renovated by the people living in the village. The internal renovation of the church has been performed during the

Figure 3.26. Overview of the colorized interior using the vertex based method. Some white balance and exposure correction has been done on the camera images.

ministry of László Mikes at the second half of the 20th century (1951-1952) and finally in the period of 2002 and 2004 the outer part of the building was renovated as well. In the same time the roof was renewed and the slate used before have been replaced by shingle respecting the strict rules regulating the renovation of the monuments. In the same time period the mechanism of the bells have been automatized, the star on the top of the tower was renovated, the pargeting was renovated as well as the doors and the windows, further the door located on the rotunda have been unfolded. In the past three years as a part of the work to keep the consistency of the building the tower as well as the roof-structure made from shingle has been repainted, the pulpit, the Chair of Moses, the gallery as well as the benches have been renovated.

The exterior model of the church of Kolozsnéma is presented in Fig. 3.28. This example illustrates well how only a reduced number of images can be sufficient to colorize the complete model of such a building: we only used 21 images in this case with good results. Of course for a more complex structure more images will be needed to cover every part without occlusion, and if important details have to be documented in higher resolution then again the number of images will increase. In this scenario, we faced a similar issue as with the organ in the sanctuary of the other church. The tombstones around the church did not allow for capturing images of the walls without occlusion, so to be able to correctly colorize the building without projecting the image of the tombstones on it, we had to make sure that

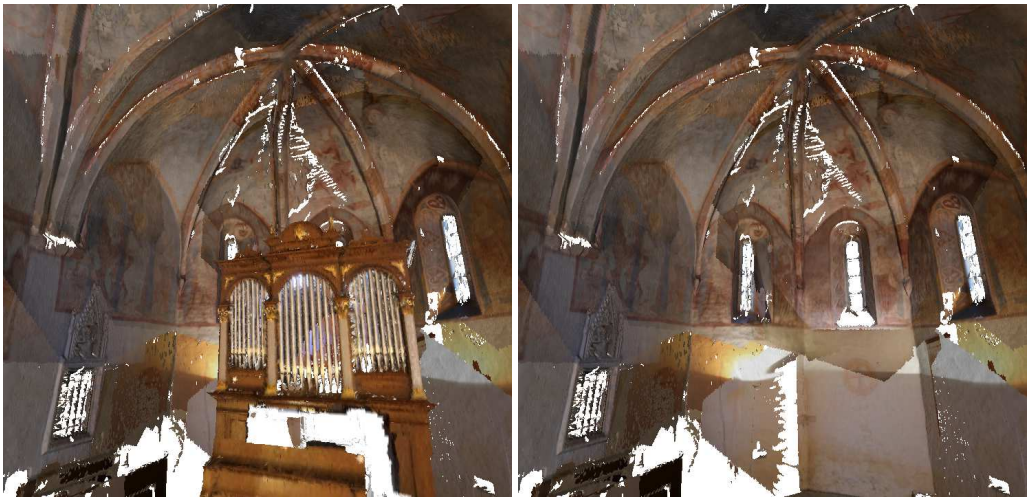

Figure 3.27. Illustrating the sanctuary with and without the organ on a textured model.

Figure 3.28. Exterior model of the Kolozsnéma church. Point cloud colorized from only 21 images using the vertex based method.

these objects are also included in the 3D scans, and we kept them in the 3D model while processing the data, simply to obstruct the parts that are not visible from a given camera's viewpoint. After the colorization is finished, these tombs can easily be removed if necessary from the model.

3.5 Summary

In this chapter, a generic, nonlinear, explicit correspondence-less pose estimation method was proposed. The absolute camera pose estimation is based on the 3D-2D registration of a common Lidar-camera planar patch. The proposed method makes use of minimal information (plain depth data from 3D and radiometric information from 2D) and is general enough to be used both for perspective and omnidirectional central cameras. The State-of-the-Art performance of the proposed method was confirmed both on large synthetic data sets as well as on various real data experiments using different depth sensors, perspective and omnidirectional cameras. The method could be further extended to handle internal camera parameter estimation as well.

Since most of the current comparable point-based methods usually heavily rely on RGB information, and struggle with surfaces having homogeneous color, or too much reflection, our method has a clear advantage by not relying on 3D color information to solve the problem. As cultural heritage is becoming a field that frequently relies on digital methods in the documenting processes, we proposed a workflow for the 2D-3D visual data fusion based on our region-based method, extended with a pose refinement step. The method enables us to fuse color-calibrated high resolution information into the 3D model, by relying on at least a single smooth region visible in both 3D and 2D domains, but it also makes use of any low quality 3D RGB information that might be available, to enhance the pose estimation results. Furthermore, since we are not directly using the RGB color values, the method works with infrared, or even with hyperspectral images, that are widely popular imaging methods amongst cultural heritage experts.

As we have found, another popular cultural heritage task is to create precise, metric, laser scanned models of buildings or excavation sites both for analyzing and reconstruction purpose. 2D-3D visual data fusion is still a major step of these processes, thus we proposed a workflow, that this time relies on the data captured and registered using the dedicated commercial software, and focuses more on the camera selection problem that is more challenging in such a large scale case. The pose estimation step of course could be replaced with the region-based method propose in this chapter, at any time. While most of the commercial solutions can give good results in the generic setup, we deal with a different, more complex case, when the camera is not attached to the scanner, this way being able to produce higher level of detail, that is necessary for the heritage applications. The proposed method chooses for every point the camera with the best view of that point based on different parameters. We also presented a texture mapping step that takes advantage of the full resolution of the captured images, and even enables us to create a reduced size model for online visualization. We have shown that the detail level of such a colorized 3D model can greatly be increased from what we might get with a camera mounted on the scanner, by capturing high resolution images of the important details by moving the camera closer and using higher focal length lenses if necessary.

Chapter 4

Planar Homography, Relative Pose and 3D Reconstruction

4.1 State of the Art Overview

Homography estimation is essential in many applications including pose estimation [106], tracking [108, 107], structure from motion [109] as well as recent robotics applications with focus on navigation [110], vision and perception [111]. Efficient homography estimation methods exist for classical perspective cameras [15], but these methods are usually not reliable in case of omnidirectional sensors. The difficulty of homography estimation with omnidirectional cameras comes from the non-linear projection model yielding shape changes in the images that make the direct use of these methods nearly impossible.

For the geometric formulation of omnidirectional cameras multiple models have been presented in Chapter 2.1. When the camera is calibrated, which is typically the case in practical application, then image points can be lifted to the surface of a unit sphere providing a unified model independent of the inner non-linear projection of the camera. The big advantage of such a generic model is that many concepts from standard projective geometry (in particular homographies or stereo triangulation techniques) remain valid for central omnidirectional cameras. For example, homography can be estimated using these spherical points [108, 107]. Of course, pose estimation must rely on the actual images taken in a real environment, hence we cannot rely on the availability of special calibration targets.

4.1.1 Related Work

Recently, region-based methods have been gaining more attention [111, 112], in particular affine invariant detectors [113]. Patch-based scene representation is proved to be efficient [114] and consistent with region-based correspondence-search methods [115]. A classical solution is to establish a set of point matches and then estimate homography based on these point pairs. For this purpose classical keypoint detectors, such as SIFT [116], are widely used [109, 107] for omnidirectional images.

Unfortunately, big variations in shape resolution and non-linear distortion challenges keypoint detectors as well as the extraction of invariant descriptors, which are key components of reliable point matching. For example, proper handling of scale-invariant feature extraction requires special considerations in case of omnidirectional sensors, yielding mathematically elegant but complex algorithms [117]. In [118] a new computation of descriptor patches was introduced for catadioptric omnidirectional cameras which also aims to

reach rotation and scale invariance. In [109], a correspondence-less algorithm is proposed to recover relative camera motion. Although matching is avoided, SIFT features are still needed because camera motion is computed by integrating over all feature pairs that satisfy the epipolar constraint. Epipolar geometry of omnidirectional camera pairs have also been studied [119], which can be used to establish dense stereo matches. A number of works discuss the possibility of featureless image matching and recognition (most notably [120]), but with limited success. Our region-based homography estimation method is not affected by any of these issues, since the strong non-linear distortion of the camera can be eliminated simply by working with normalized spherical patches instead, that are suitable for solving the homography estimation problem. We will show in this chapter that camera pose can be directly factorized from the estimated planar homography of [Frohlich, Tamas, Kato, 2016].

The importance of piecewise planar object representation in 3D stereo has been recognized by many researchers. There are various solutions in case of standard perspective cameras, many of them are making use of the plane induced homography: Habbecke and Kobbelt used a small plane, called 'disk', for surface reconstruction [122, 121]. They proved that the normal is a linear function of the camera matrix and homography. By minimizing the difference of the warped images, the surface is reconstructed. Furukawa proposed using a small patch for better correspondence [114], then The surface is grown with the expansion of the patches. The piecewise planar stereo method of Sinha *et al.* [123] uses shape from motion to generate an initial point cloud, then a best fitting plane is estimated, and finally an energy optimization problem is solved by graph cut for plane reconstruction. Combining the work by Furukawa and Sinha [114, 123], Kowdle *et al.* introduced learning and active user interaction for large planar objects [124]. Hoang *et al.* also started from a point cloud [125] which was subsequently used for creating a visibility consistent mesh. In our approach, planes are directly reconstructed from image region(s) rather than a point cloud. Fraundorfer *et al.* [126] used MSER regions to establish corresponding regions pairs. Then a homography is calculated using SIFT detector inside the regions. Planar regions are then grown until the reprojection error is small. Zhou *et al.* assumed the whole image is a planar object, and proposed a short sequence SfM framework called TRASAC [127]. The homography is calculated using optical flow. Although the role of planar regions in 3D reconstruction has been noticed by many researchers, the final reconstruction is still obtained via triangulation for most State-of-the-Art methods. Planar objects are only used for better correspondences or camera calibration. Our approach in contrast provides direct solution for the plane reconstruction problem, only relying on the planar homography estimated between image regions [Molnár *et al.*, 2014].

Multi-view 3D reconstruction also has an important role in image-based urban HDR mapping and scene reconstruction [128]. New industrial applications are gaining ground in the domain of street level mapping [129], maintenance, autonomous navigation and self localization [130]. A key component in such applications is the simultaneous and efficient solution of 3D reconstruction and pose estimation. Particularly the planar reconstruction of objects like facades, walls, tables, traffic signs is an important task in many applications. Numerous methods already exist for the extraction of *e.g.* traffic signs using CNN [131] and recognition using Deep Learning [132] or facade elements extraction using RNN and MRF [133]. Unfortunately feature-point matching on these surfaces is hard, thus classical reconstruction approaches based on sparse point correspondences [15] will struggle. However, it is well known that a planar homography between a pair of image regions contains information about both the camera relative pose and the 3D plane parameters, thus plane re-

construction is possible from such a homography [134, 15, 112]. Based on this observation, we will present a novel direct solution for the simultaneous relative pose estimation and plane reconstruction formulated through a planar homography estimation problem between corresponding image regions [Frohlich, Kato, 2018].

4.1.2 Contributions

In this chapter, multiple homography estimation algorithms are proposed, that work directly on segmented planar patches. As a consequence, our methods do not need extracted keypoints nor keypoint descriptors. In fact, we do not use any photometric information at all, hence our methods can be used even for multimodal sensors. Since segmentation is required anyway in many real-life image analysis tasks, such regions may be available or straightforward to detect. In our experiments, we have used simple interactive segmentations but automatic detection of *e.g.* windows (which are quite common planar regions in urban scenes) is also possible [135]. Furthermore, segmentation is less affected by non-linear distortions when larger blobs are extracted. The main advantage of the proposed method is the use of regions instead of point correspondences and a generic problem formulation which allows to treat several types of cameras in the same framework. We reformulate homography estimation as a shape alignment problem, which can be efficiently solved in a similar way as in [78]. We show in the first application how such a homography can be decomposed to find the relative pose of the cameras in case of a general setup, and also in case of a well known urban scene constraint, the *weak Manhattan world* assumption. Quantitative evaluation on synthetic datasets proved the methods robustness and efficiency.

Then we present a variational calculus based method for calculating the planar surface parameters in a closed form solution only from the homography estimated between spherical cameras. Quantitative evaluation on a large set of synthetic data confirms the real-time performance, efficiency and robustness of the proposed solution.

In the last application we present a direct method for simultaneous pose estimation and 3D plane reconstruction formulated as a homography estimation problem. The proposed solution works directly on segmented planar patches, and is solved in a similar way as in [78]. The main advantage of the proposed method is the generic problem formulation which allows to treat several planes and multi-view camera systems in the same framework. The method has been quantitatively evaluated on synthetic data and also on real data from the KITTI dataset.

4.2 Homography Estimation for Omni Cameras

Given a scene plane π, let us formulate the relation between its images $\mathcal{D}$ and $\mathcal{F}$ in two omnidirectional cameras represented by the unit spheres $\mathcal{S}_1$ and $\mathcal{S}_2$. The mapping of plane points $\mathbf{X}_\pi \in \pi$ to the camera spheres $\mathcal{S}_i, i = 1, 2$ is governed by (2.1), hence it is bijective (unless π is going through the camera center, in which case π is invisible). Assuming that the first camera coordinate system is the reference frame, let us denote the normal and distance of π to the origin by $\mathbf{n} = (n_1, n_2, n_3)^T$ and d, respectively. Furthermore, the relative pose of the second camera is composed of a rotation $\mathbf{R}$ and translation $\mathbf{t} = (t_1, t_2, t_3)^T$, that gives the transformation bringing a point given in the coordinate system

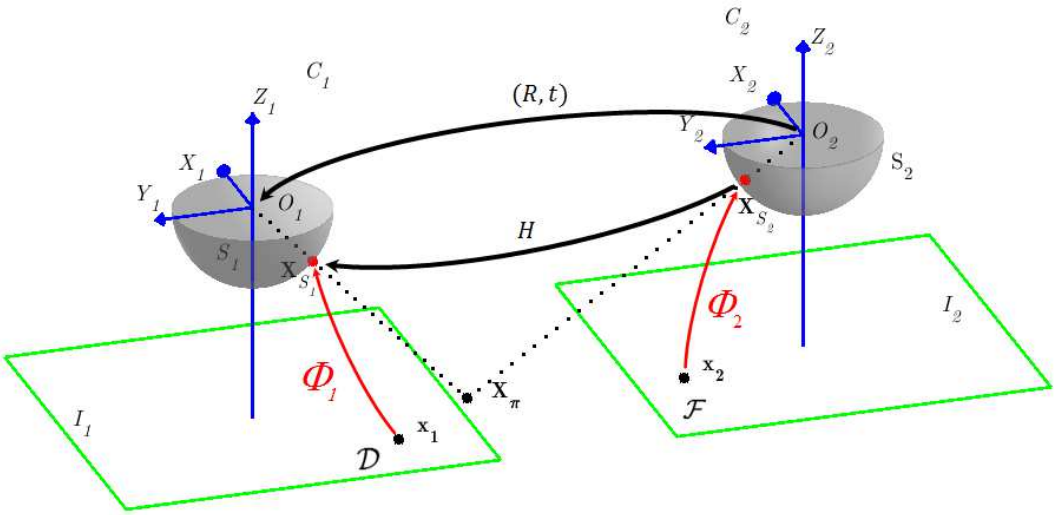

Figure 4.1. Homography acting between omnidirectional cameras represented as unit spheres.

of the second camera $\mathcal{C}_2$ into the reference frame of the first camera $\mathcal{C}_1$, as shown in Fig. 4.1

$$\mathbf{X}_{\mathcal{C}1} = \mathbf{R}\mathbf{X}_{\mathcal{C}2} + \mathbf{t}$$

thus projecting from sphere $\mathcal{S}_2$ to $\mathcal{S}_1$ is simply done by applying the same transformation, then normalizing the transformed point onto the unit sphere:

$$\mathbf{x}_{\mathcal{S}1} = \frac{\mathbf{R}\mathbf{X}_{\mathcal{S}2} + \mathbf{t}}{\|\mathbf{R}\mathbf{X}_{\mathcal{S}2} + \mathbf{t}\|}$$

Because of the single viewpoint, planar homographies, as defined in (2.12) stay valid for omnidirectional cameras too [107].

4.2.1 Planar Homography for Central Omnidirectional Cameras

From our point of view, Φ provides an equivalent *spherical image* by backprojecting the omnidirectional image onto $\mathcal{S}$ and the planar homography $\mathbf{H}$ simply acts between these spherical images [Frohlich, Tamas, Kato, 2016], as shown in Fig. 4.1. Basically, the homography transforms the rays as $\mathbf{x}_{\mathcal{S}1} \propto \mathbf{H}\mathbf{x}_{\mathcal{S}2}$, hence the transformation induced by the planar homography between the spherical points is also bijective. $\mathbf{H}$ is defined up to a scale factor, which can be fixed by choosing $h_{33} = 1$, *i.e.* dividing $\mathbf{H}$ with its last element, assuming it is non-zero. Note that $h_{33} = 0$ iff $\mathbf{H}(0,0,1)^T = (h_{13}, h_{23}, 0)^T$, *i.e.* iff the origin of the coordinate system in the first image is mapped to the ideal line in the second image. That happens only in extreme situations, *e.g.* when $Z_2 \perp Z_1$ and O_2 is on Z_1 in Fig. 4.1, which is usually excluded by physical constraints in real applications. Thus the point $\mathbf{X}_\pi$ on the plane and its spherical images $\mathbf{x}_{\mathcal{S}1}$, $\mathbf{x}_{\mathcal{S}2}$ are related by

$$\mathbf{X}_\pi = \lambda_1 \mathbf{X}_{\mathcal{S}1} = \lambda_2 \mathbf{H}\mathbf{X}_{\mathcal{S}2} \Rightarrow \mathbf{X}_{\mathcal{S}1} = \frac{\lambda_2}{\lambda_1}\mathbf{H}\mathbf{X}_{\mathcal{S}2}$$

Hence $\mathbf{X}_{\mathcal{S}1}$ and $\mathbf{H}\mathbf{X}_{\mathcal{S}2}$ are on the same ray [Frohlich, Tamas, Kato, 2016] yielding

$$\mathbf{X}_{\mathcal{S}1} = \frac{\mathbf{H}\mathbf{X}_{\mathcal{S}2}}{\|\mathbf{H}\mathbf{X}_{\mathcal{S}2}\|} \equiv \Psi(\mathbf{X}_{\mathcal{S}2}) \tag{4.1}$$

4.2.2 Homography Estimation

Given a pair of omnidirectional cameras observing a planar surface patch, how to estimate the homography between its images, the spherical regions $\mathcal{D}_{\mathcal{S}} \in \mathcal{S}_1$ and $\mathcal{F}_{\mathcal{S}} \in \mathcal{S}_2$? First, let us formulate the relation between a pair of corresponding omni image points $\mathbf{x}_1$ and $\mathbf{x}_2$. The corresponding spherical points are obtained by applying the camera's inner projection functions Φ_1, Φ_2, which are then related by (4.1):

$$\Phi_1(\mathbf{x}_1) = \mathbf{X}_{\mathcal{S}1} = \frac{\mathbf{H}\mathbf{X}_{\mathcal{S}2}}{\|\mathbf{H}\mathbf{X}_{\mathcal{S}2}\|} = \Psi(\Phi_2(\mathbf{x}_2)) \tag{4.2}$$

Any corresponding point pair $(\mathbf{x}_1, \mathbf{x}_2)$ satisfies the above equation. Thus a classical solution is to establish at least 4 such point correspondences $\{(\mathbf{x}_1^i, \mathbf{x}_2^i)\}_{i=1}^N$ by standard intensity-based point matching, and solve for $\mathbf{H}$. However, the inherent non-linear distortion of omnidirectional imaging challenges traditional keypoint detectors as well as the extraction of invariant descriptors, which are key components of reliable point matching. Therefore we are interested in a solution without finding point matches.

We will show that by identifying a single planar region in both omni images (denoted by $\mathcal{D}$ and $\mathcal{F}$, respectively), $\mathbf{H}$ can be estimated without any additional information [Frohlich, Tamas, Kato, 2016]. Since we do not have established point pairs, we cannot directly use (4.2). However, we can get rid of individual point matches by integrating both sides of (4.2) yielding a surface integral on $\mathcal{S}_1$ over the surface patches $\mathcal{D}_{\mathcal{S}} = \Phi_1(\mathcal{D})$ obtained by lifting the first omni image region $\mathcal{D}$ and $\mathcal{F}_{\mathcal{S}} = \Psi(\Phi_2(\mathcal{F}))$ obtained by lifting the second omni image region $\mathcal{F}$ and transforming it by $\Psi : \mathcal{S}_2 \to \mathcal{S}_1$. To get an explicit formula for these integrals, the surface patches $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{S}}$ can be naturally parameterized via Φ_1 and $\Psi \circ \Phi_2$ over the planar regions $\mathcal{D} \subset \mathbb{R}^2$ and $\mathcal{F} \subset \mathbb{R}^2$:

$$\begin{aligned} \forall \mathbf{X}_{\mathcal{S}_1} \in \mathcal{D}_{\mathcal{S}} \quad &: \quad \mathbf{X}_{\mathcal{S}_1} = \Phi_1(\mathbf{x}_1), \mathbf{x}_1 \in \mathcal{D} \\ \forall \mathbf{Z}_{\mathcal{S}_1} \in \mathcal{F}_{\mathcal{S}} \quad &: \quad \mathbf{Z}_{\mathcal{S}_1} = \Psi(\Phi_2(\mathbf{x}_2)), \mathbf{x}_2 \in \mathcal{F}, \end{aligned}$$

yielding the following integral equation:

$$\begin{aligned} \iint_{\mathcal{D}} \Phi_1(\mathbf{x}_1) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\, \mathrm{d}x_{12} = \\ \iint_{\mathcal{F}} \Psi(\Phi_2(\mathbf{x}_2)) \left\| \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\, \mathrm{d}x_{22} \end{aligned} \tag{4.3}$$

where the magnitude of the cross product of the partial derivatives is known as the surface element. The above integrals can be regarded as component-wise surface integrals of scalar fields, yielding a set of 2 equations. Since the value of a surface integral is independent of the parameterization, the above equality holds because both sides contain an integral on $\mathcal{S}_1$, parameterized through Φ_1 on the left hand side and through $\Psi \circ \Phi_2$ on the right hand side.

4.2.3 Construction of a System of Equations

Obviously, 2 equations are not enough to determine the 8 parameters of a homography. In order to generate more equations, let us remark that the identity relation in (4.2) remains valid when a function $\omega : \mathbb{R}^3 \to \mathbb{R}$ is acting on both sides of the equation [78]. Indeed, for a properly chosen ω

$$\omega(\mathbf{x}_{\mathcal{S}_1}) = \omega(\Psi(\Phi_2(\mathbf{x}_2))). \tag{4.4}$$

We thus obtain the following integral equation from (4.3) and (4.4)

$$\iint_{\mathcal{D}} \omega_i(\Phi_1(\mathbf{x}_1)) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\,\mathrm{d}x_{12} = \iint_{\mathcal{F}} \omega_i(\Psi(\Phi_2(\mathbf{x}_2))) \left\| \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\,\mathrm{d}x_{22} \tag{4.5}$$

The basic idea of the proposed approach is to generate sufficiently many independent equations by making use of a set of nonlinear (hence linearly independent) functions $\{\omega_i\}_{i=1}^{\ell}$. Each ω_i generates a new equation yielding a system of ℓ independent equations. Note however, that the generated equations contain no new information, they simply impose new linearly independent constraints. Although arbitrary ω_i functions could be used, power functions are computationally favorable [78]. In our experiments, we adopted the following functions:

$$\omega_i(\mathbf{x}_{\mathcal{S}}) = x_1^{l_i} x_2^{m_i} x_3^{n_i}, \quad \text{with } 0 \leq l_i, m_i, n_i \leq 2 \text{ and } l_i + m_i + n_i \leq 3 \tag{4.6}$$

These functions provide an overdetermined system of 15 equations of the form of (4.5), which can be solved in the *least squares sense* via a standard *Levenberg-Marquardt* (LM) algorithm. The solution to the system directly provides the parameters of the homography $\mathbf{H}$.

The computational complexity is largely determined by the calculation of the integrals in (4.5). Since both cameras are calibrated, Φ_1 and Φ_2 are known, hence the integrals on the left hand side are constant which need to be computed only once. However, the unknown homography $\mathbf{H}$ is involved in the right hand side through Ψ, hence these integrals have to be computed at each iteration of the LM solver. Of course, the spherical points $\mathbf{X}_{\mathcal{S}_2} = \Phi_2(\mathbf{x}_2)$ can be precomputed too, but the computation of the surface elements is more complex. First, let us rewrite the derivatives of the composite function $\Psi \circ \Phi_2$ in terms of the Jacobian $\mathbf{J}_\Psi$ of Ψ and the gradients of Φ_2:

$$\left\| \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{22}} \right\| = \left\| \mathbf{J}_\Psi \frac{\partial \Phi_2}{\partial x_{21}} \times \mathbf{J}_\Psi \frac{\partial \Phi_2}{\partial x_{22}} \right\|$$

Since the gradients of Φ_2 are independent of $\mathbf{H}$, they can also be precomputed. Hence only $\Psi(\Phi_2(\mathbf{x}_2))$ and $\mathbf{J}_\Psi(\Phi_2(\mathbf{x}_2))$ have to be calculated during the LM iterations yielding a computationally efficient algorithm [Frohlich, Tamas, Kato, 2016].

Normalization and Initialization

Since the system is solved by minimizing the algebraic error, proper normalization is critical for numerical stability [78]. Unlike in [78], spherical coordinates are already in the range

of $[-1, 1]$, therefore no further normalization is needed. However, the ω_i functions should also be normalized into $[-1, 1]$ in order to ensure a balanced contribution of each equations to the algebraic error. In our case, this can be achieved by dividing the integrals with the maximal magnitude of the surface integral over the half unit sphere. We can easily compute these integrals by parameterizing the surface via points on the unit circle in the $x - y$ plane as $f(x, y) = (x, y, \sqrt{1 - x^2 - y^2})^T, \forall \|(x, y)\| < 1$. Thus the normalizing constant N_i for the equation generated by the function ω_i is

$$N_i = \iint\limits_{\|(x,y)\|<1} |\omega_i(f(x, y))| \sqrt{\frac{1}{1 - x^2 - y^2}} \, \mathrm{d}x \, \mathrm{d}y \tag{4.7}$$

To guarantee an optimal solution, initialization is also important. In our case, a good initialization ensures that the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ overlap as much as possible. This is achieved by computing the centroids of the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ respectively, and initializing $\mathbf{H}$ as the rotation between them.

We have developed a homography estimation algorithm in this chapter, which is independent of the camera's internal projection functions Φ_1 and Φ_2. However, the knowledge of these functions as well as their gradient are necessary for the actual computation of the equations in (4.5). The pseudo code of the proposed method is presented in Algorithm 5.

Algorithm 5 The proposed homography estimation algorithm

Input: A pair of 2D omnidirectional images with the same planar region segmented
Output: Homography $\mathbf{H}$ between the spherical images of the region

1: Back-project the 2D images onto the unit spheres using Φ_1 and Φ_2.
2: Construct the system of equations of (4.5) using the polynomial ω_i functions in (4.6).
3: Normalize the equations using (4.7)
4: Initialize the homography matrix $\mathbf{H}$ with the rotation between the centroids of the shapes on the sphere.
5: Solve the normalized nonlinear system of equations using the Levenberg-Marquardt algorithm.

4.2.4 Homography Estimation Results

A quantitative evaluation of the proposed method was performed by generating a total of 9 benchmark datasets, each containing 100 image pairs. Images of 24 different shapes were used as scene planes and a pair of virtual omnidirectional cameras with random pose were used to generate the omnidirectional images of 1MP. Assuming that these 800×800 scene plane images correspond to 5×5 m patches, we place the scene plane randomly at around 1.5 m in front of the first camera with a horizontal translation of ± 1 m and $\pm[5^\circ - 10^\circ]$ rotation around all three axes. The orientation of the second camera is randomly chosen having $\pm 5^\circ$ rotation around the X and Y axis, and $\pm 10^\circ$ around the vertical Z axis, while the location of the camera center is randomly chosen from the $[45 - 55]$ cm, $[100 - 200]$ cm, and $[200 - 500]$ cm intervals, providing the first three datasets for 3 different baseline ranges. The alignment error (denoted by δ) was evaluated in terms of the percentage of non overlapping area of the omni images after applying the homography.

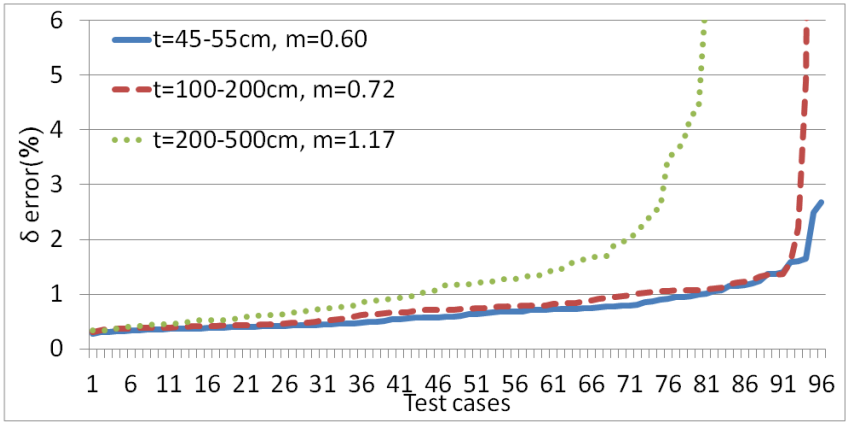

Figure 4.3. Alignment error (δ) on the synthetic dataset with various baselines (m is the median, best viewed in color).

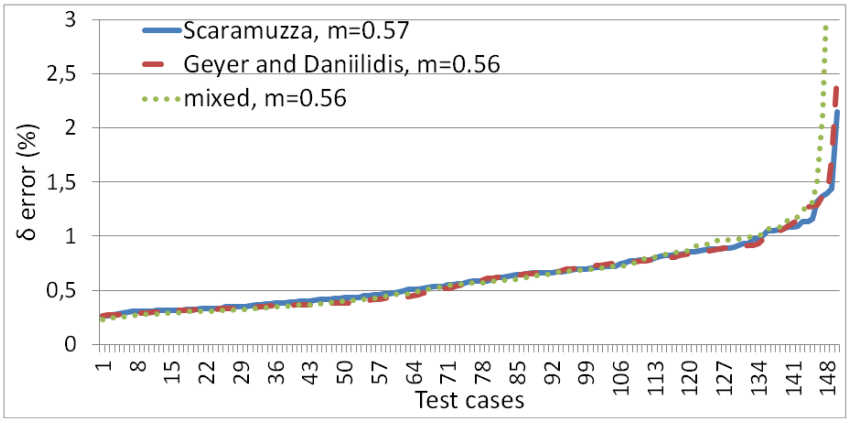

Figure 4.2. Alignment (δ) error of the homography for various internal projection models (Scaramuzza [9, 14], Geyer and Daniilidis [6], and mixed; m stands for median).

Based on our experimental results, we concluded that a δ error below 5% corresponds to a correct alignment with a visually good matching of the shapes. For the synthetic datasets, error plots are shown in Fig. 4.2, Fig. 4.3, Fig. 4.5, and Fig. 4.7. Note that each plot represents the performed test cases sorted independently in a best-to-worst sense. In Fig. 4.2, we present a quantitative comparison of homography estimation with each of the camera models described in Chapter 2.1; as well as a test case with mixed cameras, where the first camera uses the Scaramuzza's polynomial representation and the second adopts the general catadioptric model. As expected, the quality of homography estimates is independent of the internal projection functions, both models perform well, error plots almost completely overlap. Therefore in all other test cases, we will only use Scaramuzza's model from Chapter 2.1.2.

The median value of δ was 0.60%, 0.72% and 1.17% for the different baselines. In the first 2 cases, with baselines having values under 200 cm, we can say that only 1% of the results were above 5% error, while in the case of the biggest baselines $200 - 500$ cm still 84% of the results are considered good, having δ error smaller than 5%. The wrong results are typically due to extreme situations where the relative translation from the first camera to the second camera's position is in such a direction from where the image plane can be seen under a totally different angle resulting a highly different distortion of the shape on the

input image pair $\delta = 0.5\%$

input image pair $\delta = 2.6\%$

Figure 4.4. Typical registration results for the test cases with unfavorable camera pose. First row shows a test case with big translation in the z, while the second row contains a test case with region falling on the periphery of the image.

omni image.

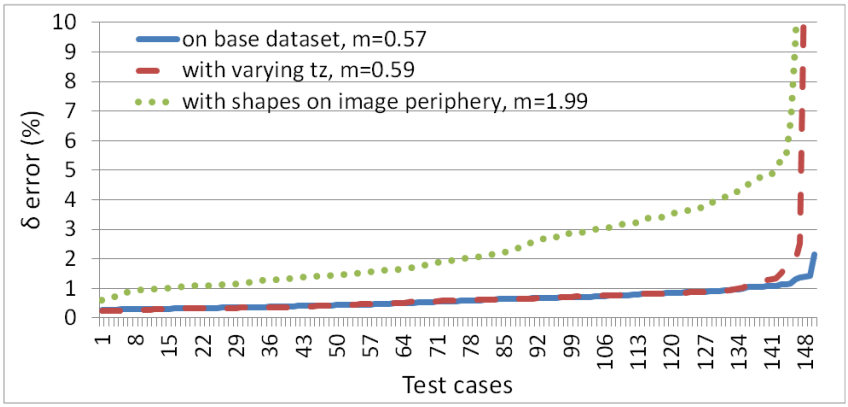

Figure 4.5. Alignment error (δ) on the synthetic datasets with unfavorable camera poses (m is the median, best viewed in color).

We have also tested the robustness of our method in some cases with unfavorable camera poses. One such situation is when the image of the actual planar region gets captured on the periphery of the omnidirectional image. It is well known, that these cameras have a much higher distortion in these regions. For this purpose we generated another synthetic dataset, making sure that all the regions fall on the periphery of the omnidirectional image. Another situation is when the relative camera pose has a much higher translation along the Z axis, resulting a considerable size difference of the regions on the omnidirectional images. For this experiment a new synthetic dataset was generated with a bigger translation along the Z axis (in the range of ± 1 m). The alignment errors of these two test cases are shown in Fig. 4.5. As we can see, the differences in the size of the regions that occur when having translation along the Z axis are well tolerated by the algorithm, a homography can be estimated with almost the same precision. On the other hand, the higher distortion at the periphery of the images results in considerable loss of resolution, hence the homography estimation also looses some precision, but the median of the δ errors are still below 2%.

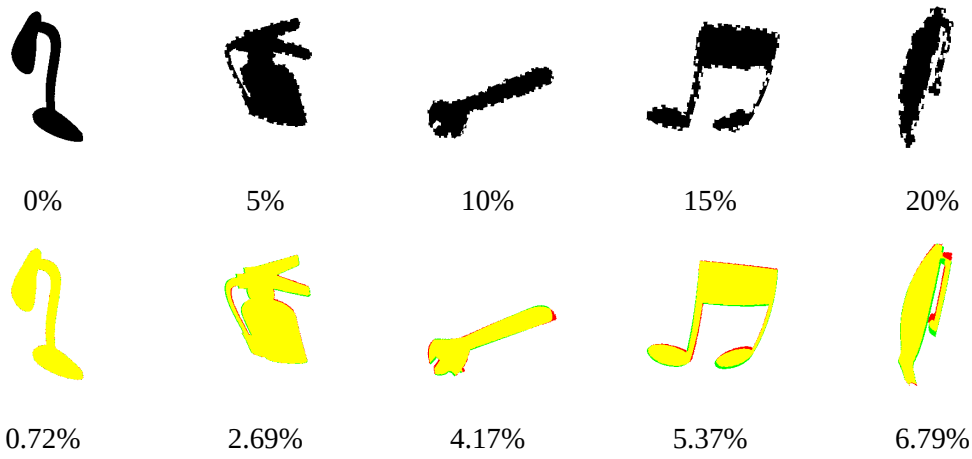

Figure 4.6. Typical registration results for various level of segmentation error. First row shows the first image and the amount of segmentation error while the second row contains the overlay of the transformed first image over the second image with the δ error (best viewed in color).

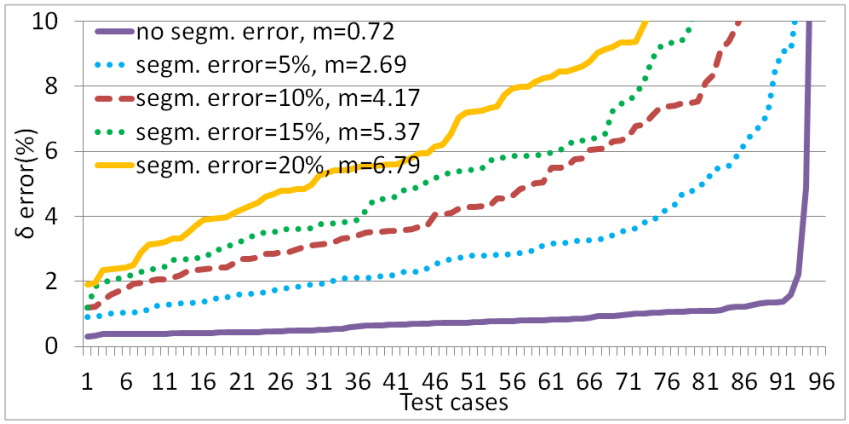

Figure 4.7. Alignment error (δ) on the synthetic dataset with various levels of boundary error (m is the median, best viewed in color).

In summary, these results demonstrate that the method is robust against both unfavorable situations.

In practice, the shapes are segmented from real world images subject to various degree of segmentation errors. Therefore robustness against segmentation errors was also evaluated on simulated data. For this we used the dataset having the typical base distances of $1-2$ m and we generated segmentation error by randomly adding and removing squares uniformly around the boundary of the shapes in one of the image pairs. A total of four datasets were produced from 5% up to 20% of boundary error. Samples from these datasets can be seen in Fig. 4.6, while Fig. 4.7 shows error plots for these datasets. Obviously, the median of δ error increases with the segmentation error, but the method shows robustness up to around 15% error level. In particular, 80% and 60% of the first two cases are visually good, while only 44% and 30% of the cases are below the desired 5% δ error for larger segmentation errors.

The algorithm was implemented in Matlab and all benchmarks were run on a standard quad-core desktop PC, resulting a typical runtime of 5 to 8 seconds without the code being

Figure 4.8. Homography estimation results on real omni image pairs. Segmented regions are overlayed in lighter color, while the result is shown on the right as the transformed green contours from the first image region over the second image.

optimized.

The real images, used for validation, were taken by a Canon 50D DSLR camera with a Canon EF 8-15mm f/4L fisheye lens and the image size was 3MP. In our experiments, segmentation was obtained by simple region growing (initialized with only a few clicks) but more sophisticated and automatic methods could also be used. The extracted binary region masks were then registered by our method and the resulting homography has been used to project one image onto the other. Three such examples are illustrated in Fig. 4.8, where the first two images are the input omni image pairs, showing the segmented region in highlight, and the third image contains the transformed edges overlayed. We can observe that in spite of segmentation errors and slight occlusions (*e.g.* by the tree in the first image of Fig. 4.8), the edges of the reprojected region and the edges on the base image are well aligned. We should also mention that while slight occlusions are well tolerated, our method does not handle the occlusion of bigger parts of the region.

In the next sections we will show two applications that rely on such estimated homographies to retrieve the relative pose of the two cameras, and even the 3D reconstruction of the planar surface used for the homography estimation.

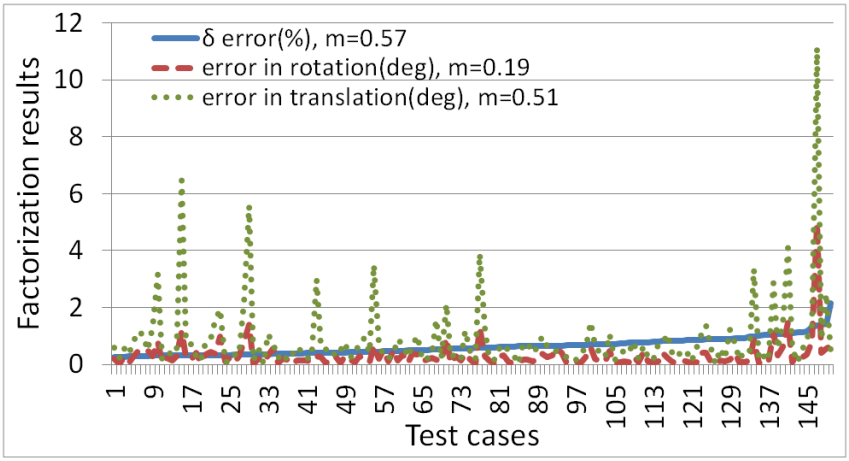

Figure 4.9. Homography factorization results showing the δ error(%) of the homography, the rotation error and the translation error as the angle between the reference and factorized translation vectors (m is the median).

4.3 Relative Pose from Homography

If we consider again that the homography $\mathbf{H}$ is composed as in (2.12) from a rotation $\mathbf{R}$, the ratio $\mathbf{t}/d$ of the translation to the distance of plane and the normal $\mathbf{n}$ of the plane, we can express the pose parameters as described in [136] using the singular value decomposition (SVD) of $\mathbf{H}$. Of course as the d distance of the plane is unknown, we can only express the translation $\mathbf{t}$ up to a scale factor. We fixed this scale factor by choosing the last element h_{33} of $\mathbf{H}$ to be 1.

The parameters that we obtain by the decomposition method can easily be verified in case of synthetic data, since we have the reference parameters saved during the dataset generation. The error in the relative translation can be characterized by either verifying the angle between the estimated and reference translation vectors, or by scaling up the estimated translation vector with the length of the reference translation and computing the Euclidean distance between them. Here we have chosen to show the former one. The results can be seen in Fig. 4.9, where test cases are sorted by increasing δ error. We can observe that on a set of 150 test cases the estimated homography is really good, the δ error was below 2% in all cases, and its median is less than 0.6%. From a good input like this, the relative rotation and translation of the cameras can be factorized with high precision, only 0.19° median error in the rotation, and 0.51° in the direction of the translation vector.

The results show, that except a few test cases, the relative pose is determined with high stability. These few test cases (the spikes on Fig. 4.9) can be better explained by looking at Fig. 4.10 which shows only the factorized pose parameters for all test cases, sorted by the rotation error. The plot confirms a clear correlation between these values, more visible on the second half of the plot, where the rotation and translation error increases together. This can be caused by the rare appearance of some specific camera configurations, where these errors in the parameters can compensate each other's effect, resulting in an overall good overlap (hence a low δ error) but spikes on Fig. 4.9.

Since the δ error of the homography in the previously mentioned dataset was considerably low (0.57% of median error), we have also tested the factorization on the datasets with simulated segmentation error used in Chapter 4.2.4, where the homography errors span on a larger scale. The rotation error can be observed in Fig. 4.11. The effect of the worse ho-

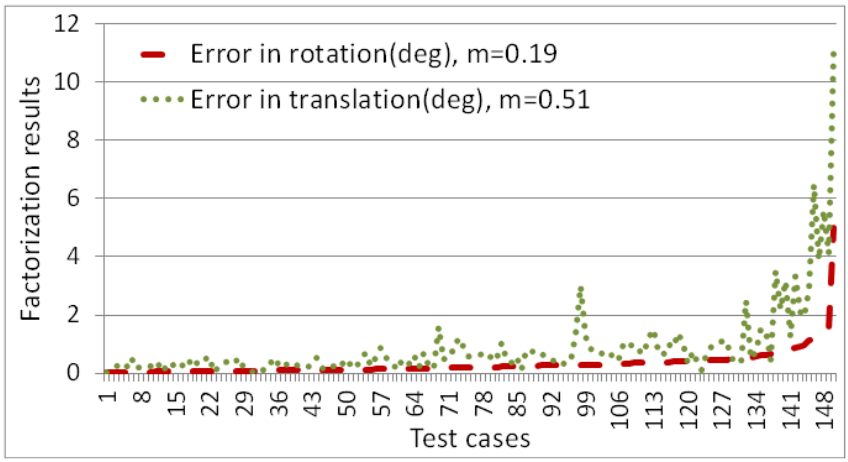

Figure 4.10. Homography factorization results showing the rotation error and the translation error as the angle between the reference and factorized translation vectors, sorted by the rotation error (m is the median).

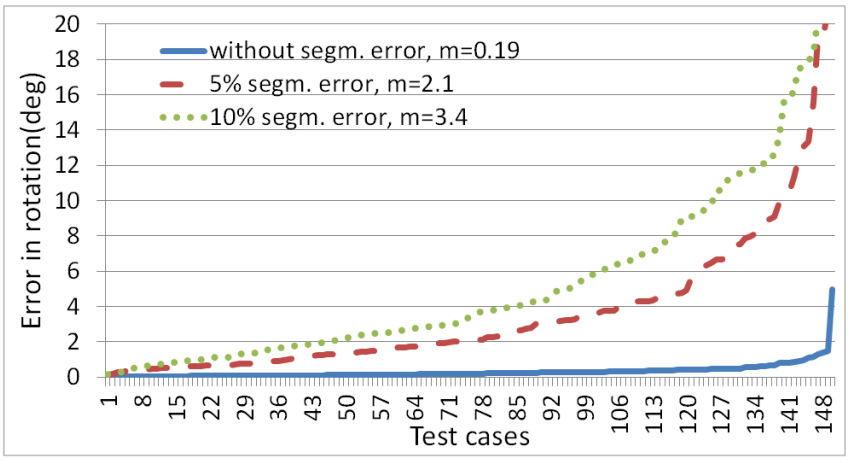

Figure 4.11. Factorized rotation error with respect to different levels of segmentation error. Test cases sorted independently (m is the median).

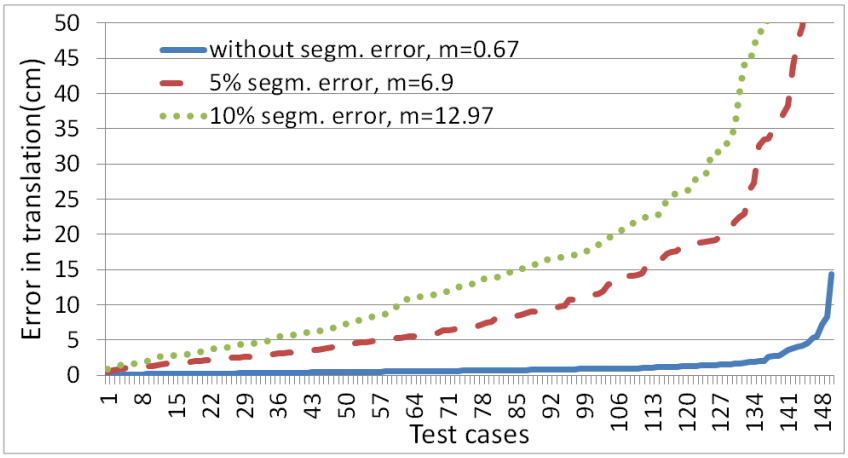

Figure 4.12. Factorized translation error with respect to different levels of segmentation error. Test cases sorted independently (m is the median).

mographies can obviously be seen on the factorized rotation, but still, at 10% segmentation error, which resulted a δ error of 4.17% for the dataset (see Fig. 4.7), the rotation error is well below 4° in median.

For the characterization of the translation errors in this case, we've expressed the Euclidean distance between the scaled up translation and the reference translation vector. The effect of the bigger δ error of the homographies in the different datasets can be observed in this case as well, visible in Fig. 4.12. The median of approximately 13 cm in the case of the 10% segmentation error can be considered a reasonably good result, since our regions represent approximately 5×5 m surfaces in the scene.

4.3.1 Manhattan World Assumption

Manhattan world assumption is quite common when working with images of urban or indoor scenes [137, 138]. Although this is a strong restriction, yet it is satisfied at least partially in man-made structures. A somewhat relaxed assumption is the *weak Manhattan world* [110] consisting of vertical planes with an arbitrary orientation but parallel to the gravity vector and orthogonal to the ground plane. Following [110], we can also take advantage of the knowledge of the vertical direction, which can be computed *e.g.* from an inertial measurement unit (IMU) attached to the camera. While [110] deals with perspective cameras, herein we will show that homographies obtained from omnidirectional cameras can also be used [Frohlich, Tamas, Kato, 2016] and then we conduct a synthetic experiment to evaluate the performance of the method.

Let us consider a vertical plane π with its normal vector $\mathbf{n} = (n_x, n_y, 0)^T$ (z is the vertical axis, see Fig. 4.1). The distance d of the plane can be set to 1, because $\mathbf{H}$ is determined up to a free scale factor. Knowing the vertical direction, the rotation matrix $\mathbf{R}$ in (2.12) can be reduced to a rotation $\mathbf{R}_z$ around the z axis, yielding

$$\begin{aligned}
\mathbf{H} &= \mathbf{R}_z - (t_x, t_y, t_z)(n_x, n_y, 0)^T \\
&= \begin{pmatrix} \cos(\alpha) - n_x t_x & -\sin(\alpha) - n_y t_x & 0 \\ \sin(\alpha) - n_x t_y & \cos(\alpha) - n_y t_y & 0 \\ n_x t_z & n_y t_z & 1 \end{pmatrix} \\
&= \begin{pmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ h_{31} & h_{32} & 1 \end{pmatrix}
\end{aligned} \tag{4.8}$$

The estimation of such a *weak Manhattan* homography matrix is done in the same way as before, but the last column of $\mathbf{H}$ is set to $(0, 0, 1)^T$, yielding 6 free parameters only [Frohlich, Tamas, Kato, 2016]. In order to quantitatively characterize the performance of our method, 2 synthetic datasets with *weak Manhattan world* assumption were generated: first the 3D scene plane is positioned vertically and randomly rotated around the vertical axis by $[-10, +10]$ degrees, followed by a translation in the horizontal direction by $\pm[400 - 800]$ pixels, equivalent to $[2 - 4]$ m such that the surface of the plane is visible from the camera. For the second camera position we used a random rotation of $[-10, +10]$ degrees around the vertical axis followed by a horizontal translation of $\pm[50 - 100]$ cm. The second dataset only differs in the vertical position of the 3D scene plane: in the first case, the plane is located approximately 150 cm higher than in the second case. Fig. 4.13 shows the registration error for these datasets. As expected, having less free parameters

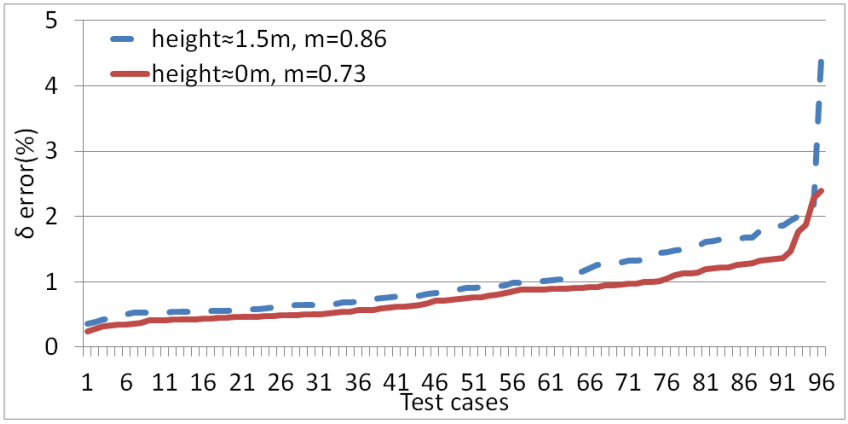

Figure 4.13. Alignment error (δ) on the synthetic dataset with *weak Manhattan constraint* (only vertical surfaces and horizontal camera rotation allowed).

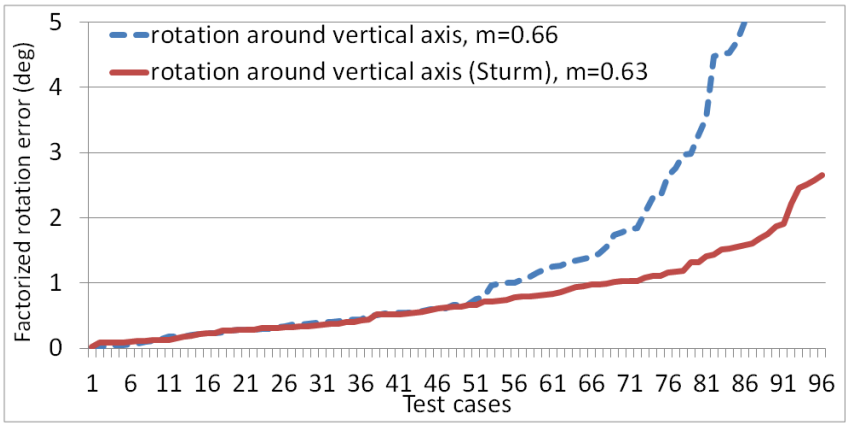

Figure 4.14. Horizontal rotation error in relative pose (m is the median).

increases estimation accuracy (alignment error is consistently under 2.5%) and decreases computational time (typically 2-3 sec.).

Based on the above parametrization, $\mathbf{H}$ can be easily decomposed in the rotation α and the translation $\mathbf{t} = (t_x, t_y, t_z)^T$ parameters of the relative motion between the cameras [Frohlich, Tamas, Kato, 2016]. For example, using the fact that $n_x^2 + n_y^2 = 1$, $t_z = \pm\sqrt{h_{31}^2 + h_{32}^2}$ (see [110] for more details).

Following the decomposition method of [110], the horizontal rotation angle of the camera can be determined with a precision of around 0.6 degrees, which means a precision of slightly above 5% of the total rotation (see Fig. 4.14). As for the translation $\mathbf{t}$, it can be also recovered with an error of less than 5 cm in the camera position. Note that the scale of $\mathbf{t}$ cannot be recovered from $\mathbf{H}$, but during the generation of our synthetic dataset we also stored the length of the translation, hence we can use it to scale up the unit direction vector obtained from $\mathbf{H}$ and compare directly the distance between the original and estimated camera centers. This is shown in the plots of Fig. 4.15.

Of course, classical homography decomposition methods could also be used. As an example, we show the pose estimation results obtained on the same dataset using the SVD-based factorization method from [106]. Fig. 4.14 and Fig. 4.15 show the rotation and translation errors for both methods. Although the differences are not big, one can clearly see the increased stability of [106].

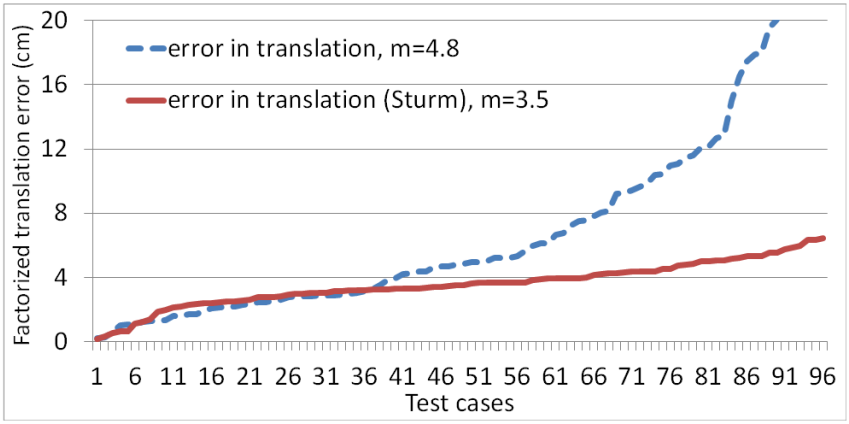

Figure 4.15. Translation error in relative pose (m is the median).

4.4 Plane Reconstruction from Homography

Returning to the equation that connects the projection rays of the two spheres through function Ψ,

$$\mathbf{X}_{\mathcal{S}1} = \frac{\mathbf{H}\mathbf{X}_{\mathcal{S}2}}{\|\mathbf{H}\mathbf{X}_{\mathcal{S}2}\|} = \Psi(\mathbf{X}_{\mathcal{S}2}) \tag{4.9}$$

we can clearly see, that the function Ψ is fully determined by the homography $\mathbf{H}$, hence estimating the homography parameters using *e.g.* the algorithm of [Frohlich, Tamas, Kato, 2016] provides the bijective mapping Ψ between the spherical points of the omnidirectional camera pair. We now present a simple, closed form solution [Molnár *et al.*, 2014] to reconstruct the normal vector of a 3D planar surface patch from the planar homography between a pair of corresponding image regions and known omnidirectional cameras, that was validated using the homography estimation method presented in Chapter 4.2. Once the normal vector $\mathbf{n}$ is determined, d can be easily computed based on (2.12) as shown *e.g.* in [15].

4.4.1 Normal Vector Computation

Although differential geometric approaches were used to solve various problems in projective 3D reconstruction, the approach proposed in [Molnár *et al.*, 2014] is unique for omnidirectional cameras to the best of our knowledge. For example, [139, 140] are about generic surface normal reconstruction using point-wise orientation- or spatial frequency disparity maps. Unlike [139, 140], which considers only projective camera and uses a parametrization dependent, non-invariant representation; [Molnár *et al.*, 2014] uses a general omnidirectional camera model and avoids point correspondences and reconstructs a planar surface from the induced planar homography between image regions.

The notations in this section are widely used in classical differential geometry. For vectors and tensors we use bold letters and italics for the coordinates. Standard basis is defined by three orthonormal vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e_3}$. 3D points $\mathbf{X} \in \mathbb{R}^3$ are identified with their coordinates in the standard basis $\mathbf{X} = X^1\mathbf{e}_1 + X^2\mathbf{e}_2 + X^3\mathbf{e}_3$ or $\mathbf{X} = X^k\mathbf{e}_k$ using the summation convention (repeated indices in superscript and subscript position mean summation). Considering the visible part of the scene object as a reasonably smooth surface $\mathbf{S}$ embedded into the ambient 3D space, $\mathbf{S}$ is represented by the general (Gauss) coordinates

u^1 and u^2 as

$$\mathbf{S}\left(u^1,u^2\right) = X^1\left(u^1,u^2\right)\mathbf{e}_1 + X^2\left(u^1,u^2\right)\mathbf{e}_2 + \\ + X^3\left(u^1,u^2\right)\mathbf{e}_3 = X^k\left(u^l\right)\mathbf{e}_k \quad (4.10)$$

The tangent space to surface $\mathbf{S}$ at a surface point (u^1,u^2) is spanned by the local (covariant) basis vectors $\mathbf{S}_k = \frac{\partial \mathbf{S}}{\partial u^k}$, $\mathbf{S}_k = \mathbf{S}_k\left(u^1,u^2\right)$, $k \in \{1,2\}$. The corresponding contravariant basis vectors $\mathbf{S}^l$, $l \in \{1,2\}$ are defined to satisfy the identity $\mathbf{S}^l \cdot \mathbf{S}_k = \delta^l_k$, where δ^l_k ,$l \in \{1,2\}$, $k \in \{1,2\}$ is the Cronecker delta and the scalar product is denoted by dot.

The normal vector of the surface is defined by the cross product $\mathbf{N} = \mathbf{S}_1 \times \mathbf{S}_2$. Surface area element is defined by the triple scalar product $|\mathbf{n}\mathbf{S}_1\mathbf{S}_2| \doteq \mathbf{n} \cdot (\mathbf{S_1} \times \mathbf{S_2})$ where $\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|}$ is the *unit normal vector* of the surface. The cross-tensor of the normal vector $\mathbf{N}_\times = \mathbf{S}_2\mathbf{S}_1 - \mathbf{S}_1\mathbf{S}_2$ is a difference of two dyadic products of the local basis vectors. Dyadic (direct) products are denoted by a simple sequence of the constituent vectors.

The dot product between dyads and vectors is defined such that $\mathbf{uv} \cdot \mathbf{w} = (\mathbf{v} \cdot \mathbf{w})\,\mathbf{u}$. From this, using the triple product expansion formula $\mathbf{N}_\times \cdot \mathbf{v} = \mathbf{N} \times \mathbf{v}$ for any vector $\mathbf{v}$.

As usual, for the representation of vectors and second order tensors purely with their coordinates we use row vectors and two dimensional matrixes. The coordinate representation of a non-scalar quantity $\mathbf{Q}$ is denoted by $[\mathbf{Q}]$.

Camera Model Independent Correspondence Equations

Let us now have a closer look at the relation between a 3D point $\mathbf{X}$ and its 2D images (x^1_i, x^2_i) and (x^1_j, x^2_j) in a pair of cameras i and j. This has been studied in [141] for establishing an affine transformation between the images of a known surface using known projection functions. First we briefly overview the derivation of this relation and then we will show how to use it for computing normal vectors of planar surface patches from corresponding image regions.

An image of the scene is basically a 3D→2D mapping given by two smooth projection functions, the so called coordinate functions: $x^1\left(X^1,X^2,X^3\right)$ and $x^2\left(X^1,X^2,X^3\right)$ with (x^1,x^2) being the 2D image coordinates. [Molnár *et al.*, 2014] doesn't assume any special form of these coordinate-functions except their differentiability w.r.t. the spatial coordinates X^1, X^2, X^3. If the projected points are on the surface (4.10) too, the image coordinates depend on the general parameters as well:

$$x^1 = x^1\left(X^1\left(u^1,u^2\right), X^2\left(u^1,u^2\right), X^3\left(u^1,u^2\right)\right) \\ x^2 = x^2\left(X^1\left(u^1,u^2\right), X^2\left(u^1,u^2\right), X^3\left(u^1,u^2\right)\right) \quad (4.11)$$

The mapping in (4.11) can be considered bijective in a small open disk around the point (u^1,u^2). Assuming that both the projection functions and the surface are smooth, these are the conditions for differentiability and local invertibility. The differential $[d\mathbf{u}] = \begin{bmatrix} du^1 & du^2 \end{bmatrix}^T$ represents a point shift on the surface with its effect on the image being $d\mathbf{x} \approx \mathbf{J} \cdot d\mathbf{u}$ where $[d\mathbf{x}] = \begin{bmatrix} dx^1 & dx^2 \end{bmatrix}^T$ and the Jacobian $\mathbf{J}$ of the mapping is invertible [Molnár *et al.*, 2014].

Now consider a camera pair, distinguishing them with indices i and j (note that i, j indices used in subscript position doesn't stand for "covariant" quantities). Since $\mathbf{J}_i$ is

invertible, we can establish correspondences between the images taking the same point shift $d\mathbf{u} \approx \mathbf{J}_i \cdot d\mathbf{x}_i$:

$$d\mathbf{x}_j = \mathbf{J}_j \cdot \mathbf{J}_i^{-1} \cdot d\mathbf{x}_j = \mathbf{J}_{ij} \cdot d\mathbf{x}_j \tag{4.12}$$

where $\mathbf{J}_{ij}$ is the Jacobian of the $\mathbf{x}_i \rightarrow \mathbf{x}_j$ mapping. Now consider the derivative of a composite function $f\left(X^l\left(u^k\right)\right)$, $l \in \{1,2,3\}$, $k \in \{1,2\}$:

$$\frac{\partial f}{\partial u^k} = \frac{\partial X^l}{\partial u^k}\frac{\partial f}{\partial X^l} = \mathbf{S}_k \cdot \nabla f, \tag{4.13}$$

where ∇f is the gradient w.r.t. the spatial coordinates and $\mathbf{S}_k$ is the local basis vector along the parameter line u^k. Applying this result to the projection functions, the components of the Jacobians take the following form [Molnár *et al.*, 2014]:

$$\begin{aligned}
[\mathbf{J}_i] &= \begin{bmatrix} \mathbf{S}_1 \cdot \nabla x_i^1 & \mathbf{S}_2 \cdot \nabla x_i^1 \\ \mathbf{S}_1 \cdot \nabla x_i^2 & \mathbf{S}_2 \cdot \nabla x_i^2 \end{bmatrix}, \\
[\mathbf{J}_j] &= \begin{bmatrix} \mathbf{S}_1 \cdot \nabla x_j^1 & \mathbf{S}_2 \cdot \nabla x_j^1 \\ \mathbf{S}_1 \cdot \nabla x_j^2 & \mathbf{S}_2 \cdot \nabla x_j^2 \end{bmatrix}
\end{aligned} \tag{4.14}$$

Substituting (4.14) into (4.12), the products of the components of (4.14) enter into $\mathbf{J}_{ij}$. For example, the determinant becomes

$$\det[\mathbf{J}_i] = \nabla x_i^1 \cdot \left(S_1 \cdot \nabla x_i^1\right)\left(S_2 \cdot \nabla x_i^2\right) - \left(S_2 \cdot \nabla x_i^1\right)\left(S_1 \cdot \nabla x_i^2\right) \tag{4.15}$$

which can be expressed by dyadic products equivalent to the surface normal's cross tensor as

$$\begin{aligned}
\det[\mathbf{J}_i] &= \nabla x_i^1 \cdot (S_1 S_2 - S_2 S_1) \cdot \nabla x_i^2 \\
&= -\nabla x_i^1 \cdot \mathbf{N}_\times \cdot \nabla x_i^2 = -|\mathbf{N}| \left|\nabla x_i^1 \mathbf{n} \nabla x_i^2\right|,
\end{aligned} \tag{4.16}$$

where $|\mathbf{N}|$ is the absolute value (length) of the surface normal vector [Molnár *et al.*, 2014]. The components of the Jacobian $\mathbf{J}_{ij}$ are then [141]:

$$[\mathbf{J}_{ij}] = \frac{1}{|\nabla x_i^1 \mathbf{n} \nabla x_i^2|}\begin{bmatrix} |\nabla x_j^1 \mathbf{n} \nabla x_i^2| & |\nabla x_i^1 \mathbf{n} \nabla x_j^1| \\ |\nabla x_j^2 \mathbf{n} \nabla x_i^2| & |\nabla x_i^1 \mathbf{n} \nabla x_j^2| \end{bmatrix} \tag{4.17}$$

The above quantities are all invariant first-order differentials: the gradients of the projections and the surface unit normal vector. Note that (4.17) is a general formula: neither a special form of projections, nor a specific surface is assumed here, hence it can be applied for any camera type and for any reasonably smooth surface.

In [Molnár *et al.*, 2014] it was shown how to use the above formula for computing the normal vector $\mathbf{n}$, when both the projection functions and the Jacobian $\mathbf{J}_{ij}$ are known. Let us write the matrix components estimated either directly with affine estimator or taking the derivatives of an estimated planar homography [1] as:

$$[\mathbf{J}_{ij}]_{est} = \begin{bmatrix} a_1^1 & a_2^1 \\ a_1^2 & a_2^2 \end{bmatrix} \tag{4.18}$$

To eliminate the common denominator we can use ratios, which can be constructed

[1] The derivatives of a planar homography provides exact affine components.

using either row, column, or cross ratios [Molnár *et al.*, 2014]. Without loss of generality, the equation for the 3D surface normal can be deduced using cross ratios $\frac{a_1^1}{a_2^2}$ and $\frac{a_2^1}{a_1^2}$. After rearranging equation $[\mathbf{J}_{ij}]_{est} = [\mathbf{J}_{ij}]$ we obtain:

$$\begin{aligned}\mathbf{n} \cdot \left[a_2^2 \left(\nabla x_i^2 \times \nabla x_j^1\right) - a_1^1 \left(\nabla x_j^2 \times \nabla x_i^1\right)\right] &= 0 \\ \mathbf{n} \cdot \left[a_1^2 \left(\nabla x_j^1 \times \nabla x_i^1\right) - a_2^1 \left(\nabla x_i^2 \times \nabla x_j^2\right)\right] &= 0\end{aligned} \tag{4.19}$$

Here we have two (known) vectors, both perpendicular to the normal:

$$\begin{aligned}\mathbf{p} &= \mathbf{n} \cdot \left[a_2^2 \left(\nabla x_i^2 \times \nabla x_j^1\right) - a_1^1 \left(\nabla x_j^2 \times \nabla x_i^1\right)\right] \\ \mathbf{q} &= \mathbf{n} \cdot \left[a_1^2 \left(\nabla x_j^1 \times \nabla x_i^1\right) - a_2^1 \left(\nabla x_i^2 \times \nabla x_j^2\right)\right]\end{aligned} \tag{4.20}$$

Thus the surface normal can readily be computed as

$$\mathbf{n} = \frac{\mathbf{p} \times \mathbf{q}}{|\mathbf{p} \times \mathbf{q}|}. \tag{4.21}$$

In the remaining part of this section, we will show based on [Molnár *et al.*, 2014], how to compute the coordinate gradients $\nabla x_k^l, k = i, j; l = 1, 2$ w.r.t. spatial coordinates and $\mathbf{J}_{ij}$ in (4.17) for an omnidirectional camera pair.

Computing Coordinate Gradients for the Spherical Camera Model

The Jacobian (4.17) includes the coordinate gradients w.r.t. spatial coordinates. These quantities were derived in [Molnár *et al.*, 2014] for the general spherical camera model presented in Chapter 2.1.2. For the sake of simplicity, the calculations are done in the camera coordinate system, but coordinate gradients calculated below can be easily transformed into any world coordinate system by applying the rotation between that world coordinate frame and the camera.[2]

As described in Chapter 2.1.2, the function Φ is fully defined by the internal camera parameters (a_0, a_2, a_3, a_4). Therefore the unit projection sphere $\mathcal{S}$ can be naturally parametrized by the omni image coordinates $\mathbf{x} = (x^1, x^2)$. Spatial points $\mathbf{X} \in \mathbb{R}^3$ are identified by the unit sphere points (*i.e.* the directions) denoted by $\mathbf{X}_\mathcal{S}$, where $\mathbf{X}_\mathcal{S} \cdot \mathbf{X}_\mathcal{S} \equiv 1$, and their distance from the projection sphere's center denoted by $x^3 \equiv \|\mathbf{X}\|$ such that

$$\mathbf{X} = x^3 \mathbf{X}_\mathcal{S}. \tag{4.22}$$

Note that the above equation follows from (2.1) and it is a non-Cartesian parametrization of $\mathbb{R}^3$ from which the gradients of the first two parameters (x^1, x^2) are required. The identity

$$\delta_k^l = \frac{\partial \mathbf{X}}{\partial x^k} \cdot \frac{\partial x^l}{\partial \mathbf{X}} = \mathbf{g}_k \cdot \nabla x^l \tag{4.23}$$

is the basic differential geometry relation between the covariant $\mathbf{g}_k = \frac{\partial X}{\partial x^k}$ and contravariant $\nabla x^l = \mathbf{g}^l$ basis vectors of the parametrization [Molnár *et al.*, 2014]. Applying (4.23) to

[2] Gradients are constructed by derivation, hence the translation to any other world coordinate system cancels out from the formulae.

(4.22), we have:

$$\begin{aligned}\mathbf{g}_k &= \frac{\partial \mathbf{X}}{\partial x^k} = x^3 \frac{\partial \Phi}{\partial x^k},\ k \in \{1,2\} \\ \mathbf{g}_3 &= \frac{\partial \mathbf{X}}{\partial x^3} = \mathbf{X}_{\mathcal{S}}\,.\end{aligned} \tag{4.24}$$

From this, the metric tensor components $g_{kl} = \mathbf{g}_k \cdot \mathbf{g}_l$, $k,l \in \{1,2,3\}$ are

$$\begin{aligned} g_{kl} = g_{lk} &= \left(x^3\right)^2 \frac{\partial \Phi}{\partial x^k} \cdot \frac{\partial \Phi}{\partial x^l},\ k,l \in \{1,2\} \\ g_{k3} = g_{3k} &= 0,\ k \in \{1,2\} \\ g_{33} &= \mathbf{X}_{\mathcal{S}} \cdot \mathbf{X}_{\mathcal{S}} = 1\,. \end{aligned} \tag{4.25}$$

Note that the second line of (4.25) follows from the derivation of the constraint $\mathbf{X}_{\mathcal{S}} \cdot \mathbf{X}_{\mathcal{S}} \equiv 1$. Using the basic result from differential geometry $\mathbf{g}^l = g^{lk}\mathbf{g}_k$, where g^{lk} are the components of the inverse metric tensor, and observing that the metric tensor has the special form $\begin{bmatrix} [g_{lk}] & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$, the first two contravariant basis vectors (the sought coordinate gradients) can be independently expressed [Molnár *et al.*, 2014] from the third vector such that

$$\begin{aligned} \begin{bmatrix} \nabla x^1 \\ \nabla x^2 \end{bmatrix} &= \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} \\ &= \frac{1}{x^3} \begin{bmatrix} \frac{\partial \Phi}{\partial x^1} \cdot \frac{\partial \Phi}{\partial x^1} & \frac{\partial \Phi}{\partial x^1} \cdot \frac{\partial \Phi}{\partial x^2} \\ \frac{\partial \Phi}{\partial x^1} \cdot \frac{\partial \Phi}{\partial x^2} & \frac{\partial \Phi}{\partial x^2} \cdot \frac{\partial \Phi}{\partial x^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \Phi}{\partial x^1} \\ \frac{\partial \Phi}{\partial x^2} \end{bmatrix}. \end{aligned} \tag{4.26}$$

In the above equation, coordinate gradients are expressed purely with the unit sphere's local basis vectors $\widetilde{\mathbf{g}}_k = \frac{\partial \Phi}{\partial x^k}$ induced by the image coordinates and the distance between the observed point and the center of the projection sphere x^3. Note that x^3 cancels out from the normal calculation in (4.21) by division. Once the normal is determined, any component of (4.17) provides an equation for $\frac{x_i^3}{x_j^3}$.

Computing the Jacobian Components

Let us now see how to construct the elements a_l^k of the Jacobian matrix in (4.18) acting directly between the omnidirectional images. Denoting the Cartesian coordinates w.r.t. the centers of the unit spheres representing the cameras i and j by $[\mathbf{x}_i] = \begin{bmatrix} z_i^1 & z_i^2 & z_i^3 \end{bmatrix}^T$ and $[\mathbf{x}_j] = \begin{bmatrix} z_j^1 & z_j^2 & z_j^3 \end{bmatrix}^T$. These spherical points are related by the bijective mapping Ψ as derived in Chapter 4.2.1, which can be directly estimated by estimating the homography between the cameras, *e.g.* with the method presented in Chapter 4.2.2. Its Jacobian $\mathbf{J}_\Psi$, composed of the partial derivatives $h_l^k \doteq \frac{\partial z_j^k}{\partial z_i^l}$, associates coordinate differentials from the sphere points i to the sphere points j:

$$\begin{bmatrix} dz_j^1 \\ dz_j^2 \\ dz_j^3 \end{bmatrix} = \begin{bmatrix} h_1^1 & h_2^1 & h_3^1 \\ h_1^2 & h_2^2 & h_3^2 \\ h_1^3 & h_2^3 & h_3^3 \end{bmatrix} \begin{bmatrix} dz_i^1 \\ dz_i^2 \\ dz_i^3 \end{bmatrix} \tag{4.27}$$

This Jacobian can be translated to the Jacobian that acts between image coordinates x_j^k and x_i^l, $k, l \in \{1, 2\}$. According to [Molnár *et al.*, 2014], the condition expressing that two nearby points are constrained to a sphere can be written as

$$\left(z^1 + dz^1\right)^2 + \left(z^2 + dz^2\right)^2 + \left(z^1 + dz^3\right)^2 = \left(z^1\right)^2 + \left(z^2\right)^2 + \left(z^3\right)^2 , \tag{4.28}$$

hence

$$z^1 dz^1 + z^2 dz^2 + z^3 dz^3 = 0 . \tag{4.29}$$

From (4.29), the third differential is

$$dz^3 = -\left(\frac{z^1}{z^3} dz^1 + \frac{z^2}{z^3} dz^2\right) . \tag{4.30}$$

This differential constraint reduces the DoF of the Jacobian in (4.27) by one. Only two lines remain linearly independent. Choosing the first two lines and replacing dz_i^3 by the right hand side of (4.30), the equations between the coordinate differentials become

$$\begin{bmatrix} dz_j^1 \\ dz_j^2 \end{bmatrix} = \begin{bmatrix} h_1^1 - \frac{z_i^1}{z_i^3} h_3^1 & h_2^1 - \frac{z_i^2}{z_i^3} h_3^1 \\ h_1^2 - \frac{z_i^1}{z_i^3} h_3^2 & h_2^2 - \frac{z_i^2}{z_i^3} h_3^2 \end{bmatrix} \begin{bmatrix} dz_i^1 \\ dz_i^2 \end{bmatrix} . \tag{4.31}$$

According to (2.6), image points x^l, $l \in \{1, 2\}$ and sphere points z^k, $k \in \{1, 2\}$ are related by the bijective mapping Φ on the whole domain of estimation. Therefore the differentials are related by

$$\begin{bmatrix} dz^1 \\ dz^2 \end{bmatrix} = \begin{bmatrix} \frac{\partial z^1}{\partial x^1} & \frac{\partial z^1}{\partial x^2} \\ \frac{\partial z^2}{\partial x^1} & \frac{\partial z^2}{\partial x^2} \end{bmatrix} \begin{bmatrix} dx^1 \\ dx^2 \end{bmatrix} ,$$

hence the Jacobian that maps image differentials $d\mathbf{x}_j = \mathbf{J}_{ij} \cdot d\mathbf{x}_j$ is as follows:

$$[\mathbf{J}_{ij}] = \begin{bmatrix} \frac{\partial \Phi_j^1}{\partial x_j^1} & \frac{\partial \Phi_j^1}{\partial x_j^2} \\ \frac{\partial \Phi_j^2}{\partial x_j^1} & \frac{\partial \Phi_j^2}{\partial x_j^2} \end{bmatrix}^{-1} \begin{bmatrix} h_1^1 - \frac{\Phi_i^1}{\Phi_i^3} h_3^1 & h_2^1 - \frac{\Phi_i^2}{\Phi_i^3} h_3^1 \\ h_1^2 - \frac{\Phi_i^1}{\Phi_i^3} h_3^2 & h_2^2 - \frac{\Phi_i^2}{\Phi_i^3} h_3^2 \end{bmatrix} \begin{bmatrix} \frac{\partial \Phi_i^1}{\partial x_i^1} & \frac{\partial \Phi_i^1}{\partial x_i^2} \\ \frac{\partial \Phi_i^2}{\partial x_i^1} & \frac{\partial \Phi_i^2}{\partial x_i^2} \end{bmatrix} . \tag{4.32}$$

Like the coordinate gradients, (4.32) contains only the components of unit spheres' local basis vectors $\frac{\partial \Phi_i}{\partial x_i^k}$ $k \in \{1, 2\}$ and $\frac{\partial \Phi_j}{\partial x_j^l}$ $l \in \{1, 2\}$. Since both cameras are calibrated, Φ_i and Φ_j are known. Furthermore, the homography $\mathbf{H}$ acting between the (spherical) regions $\mathcal{D}$ and $\mathcal{F}$ corresponding to the scene plane π has been computed using [Frohlich, Tamas, Kato, 2016], Ψ is also know, hence $\mathbf{J}_{ij}$ is fully determined.

In summary, given a pair of corresponding regions F and D in a pair of calibrated omnidirectional cameras with known projection functions Φ_i, Φ_j, the 3D scene plane π can be reconstructed through the following steps:

1. Estimate the homography $\mathbf{H}$ acting between the corresponding spherical regions $\mathcal{F}$ and $\mathcal{D}$ (using *e.g.* [Frohlich, Tamas, Kato, 2016]), which gives Ψ.

2. Estimate the relative pose $(\mathbf{R}, \mathbf{t})$ between the cameras. Given $\mathbf{H}$, this can be done by homography factorization method, *e.g.* [106], [Frohlich, Tamas, Kato, 2016].

3. Compute the normal $\mathbf{n}$ of π using the direct formula (4.21), and then d by a standard

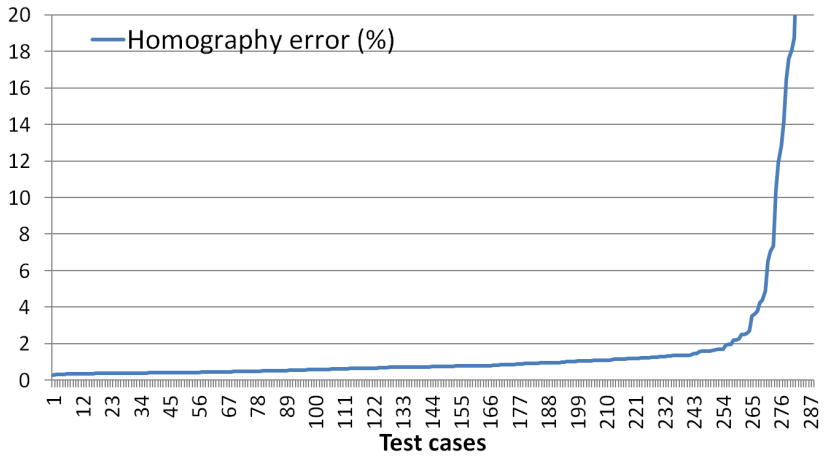

Figure 4.16. Homography error for the synthetic datasets (test cases sorted by δ error).

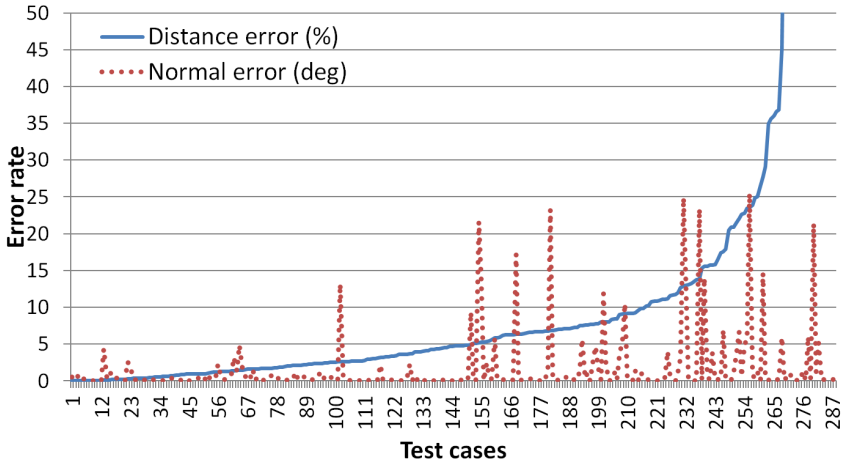

Figure 4.17. Distance error and normal error plot for the synthetic datasets (test cases sorted on the x axis based on distance error, normal error values are scaled with the factor of 0.3 for better visualization).

method based on (2.12) [15].

4.4.2 Reconstruction Results

The proposed method was tested on 3 datasets, each having approximately 100 image pairs. Images of 24 different shapes were used as scene planes and a pair of virtual omnidirectional cameras with random pose were used to generate the omni image pairs. Assuming that a 800×800 pixels scene corresponds to a 5×5 m patch, we positioned the virtual cameras at distances from the $45 - 55$ cm, $100 - 200$ cm, and $200 - 500$ cm intervals respectively, resulting 3 datasets with different camera base distances. The first step of our algorithm is estimating a homography between the omnidirectional cameras. For this purpose, we use the region-based method presented in Chapter 4.2. For reference, we show the homography error on our synthetic dataset in terms of the percentage of non overlapping area (δ error) sorted in increasing order in Fig. 4.16. The produced homographies have less than 2% error for about 256 examples. This is important as it directly affects the reconstruction accuracy of our method.

Once the planar homography between the corresponding region pair is estimated, we

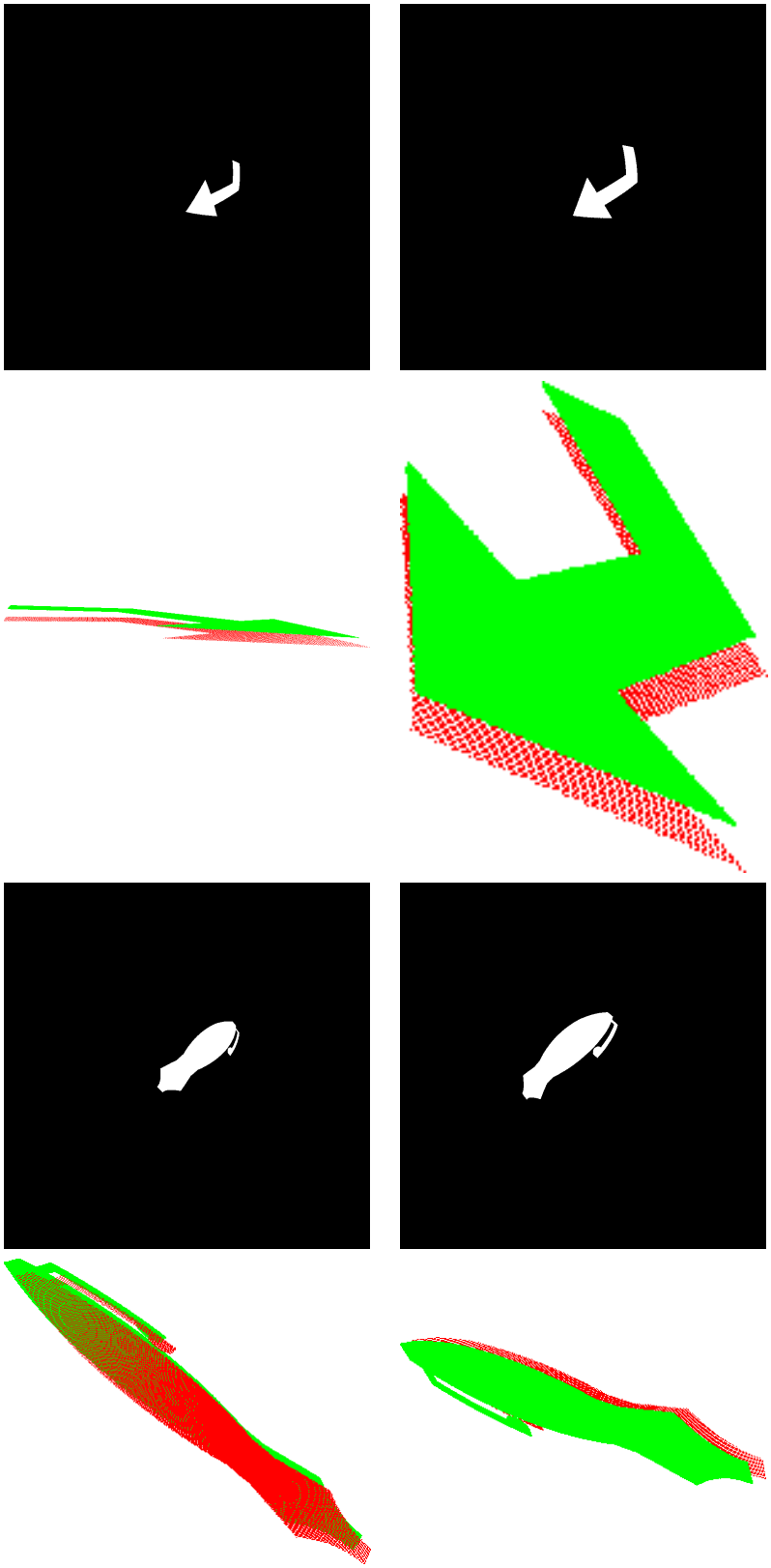

Figure 4.18. Reconstruction results from a pair of synthetic omni images (red: reconstructed, green: original 3D planar patch)

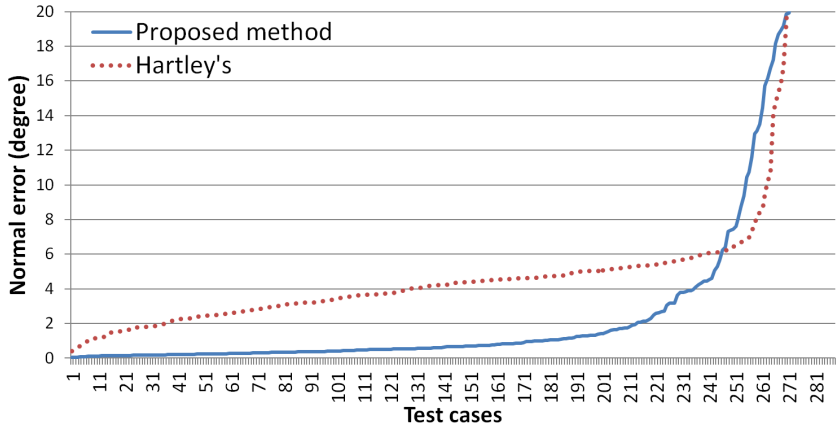

Figure 4.19. Comparative normal error plot on our synthetic dataset with the method from [15] (test cases sorted independently for the two methods)

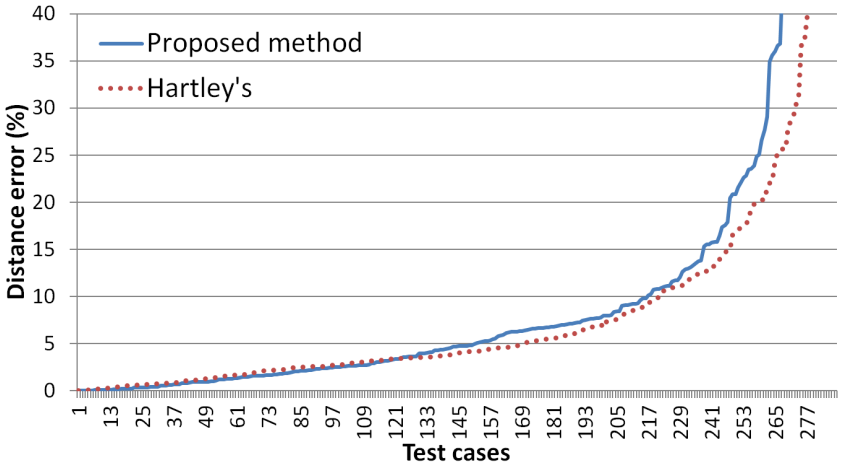

Figure 4.20. Comparative distance error plot on our synthetic dataset with the method from [15] (test cases sorted independently for the two methods).

can compute the 3D surface normal and distance using the proposed closed form formula. Sample 3D reconstructions for synthetic data is shown in Fig. 4.18. The green surface is the ground truth surface and the red one is the recovered surface. Fig. 4.17 shows the correlation of the error plots for the whole synthetic dataset. It is clear that distance error plot runs together with the normal error, hence our method provides reliable reconstructions for most test cases, giving low error rates for both surface parameters.

It is worth mentioning that the reconstruction algorithm's runtime is only 8 ms running in Matlab on an Intel i7 3.4 GHz CPU with 8GB memory. This means it can reach real-time speed due to the closed form solution adopted.

Comparison with a Classical Solution

We have performed an experimental comparison of our method with a well known classical plane from homography method described by Hartley and Zisserman [15] (the Matlab code used is vgg_plane_from_2P_H.m[3]) and quantitatively demonstrated the performance of our method with respect to that algorithm. The purpose of this experiment is to compare

[3] `http://www.robots.ox.ac.uk/~vgg/hzbook/code/`

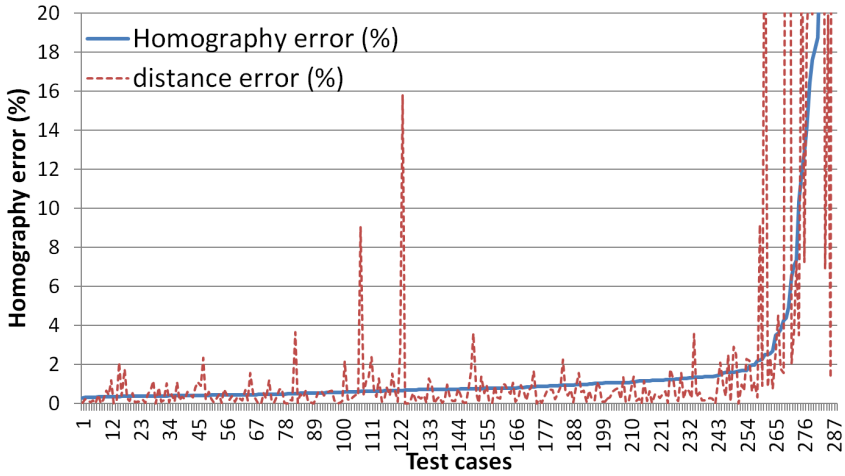

Figure 4.21. Distance error rates (scaled with a factor of 0.1 for better visualization) corresponding to the homography error (test cases sorted by the homography error)

our direct method derived via differential geometric considerations with a classical direct methods derived via projective geometric considerations, as a basis. Results show that our method is significantly better in determining the correct normal vector. The error shown in Fig. 4.19 is computed as the angle in degrees between the calculated and the ground truth normal vectors: mean value of our method was only 0.66°, while the classical plane from homography method produced 4.32° error on average. We remark that an error above 5° can be considered a completely wrong result. The relative distance error of the reconstructed plane is shown in Fig. 4.20. On these plots we can see that the precision of the two methods is almost identical, because both approaches use a similar way to compute d, giving a mean value of 4.0% and 4.7% respectively, Hartley's being the better.

Robustness

As we mentioned before, the precision of the estimated homography is crucial for 3D reconstruction. As we can see in Fig. 4.21 the distance error of the reconstruction is low, until the homography error is below $2-3\%$ but then with bigger homography error it increases exponentially. We can observe the same behavior in the normal vector calculation as shown in Fig. 4.22.

The accuracy of the proposed method depends not only on the quality of the homography estimation, but also on the determined camera pose parameters. Obviously, normal estimation is only affected by the rotation matrix, while distance calculation depends on both rotation and translation. To characterize the robustness of our method against errors in these parameters, we added various percent of noise to the original values and quantitatively evaluated the reconstruction error on our synthetic dataset. Table 4.1 and Table 4.2 show that both distance and normal estimation are sensitive to rotation errors in the camera pose, being robust up to 2° degree of rotation error, and distance estimation can tolerate up to 5% translation error as well (see Table 4.3). Normal estimation is more sensitive to rotation error around the Z axis, while distance errors increase more with rotation errors around the X axis.

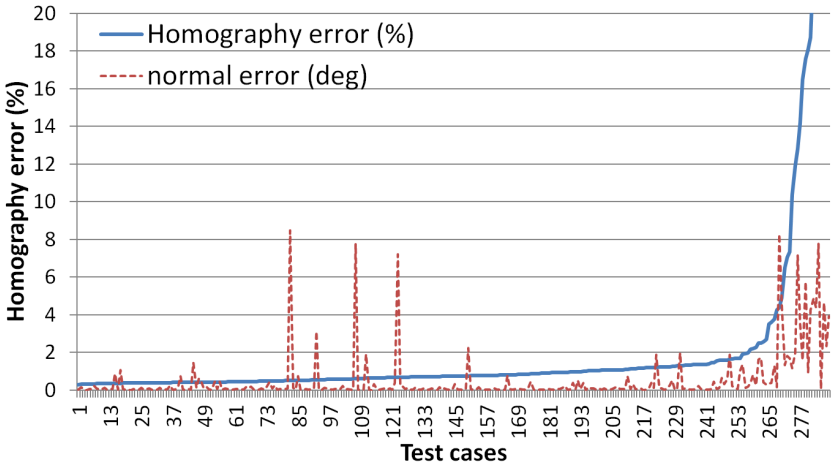

Figure 4.22. Normal error rates (scaled with a factor of 0.1 for better visualization) corresponding to the homography error (test cases sorted by the homography error)

Table 4.1. Normal error(deg) w.r.t. rotation error added in different axes

Noise(deg)	0	0.5	1	2	4
x	0.55	0.85	1.46	1.89	4.14
y	0.55	0.78	1.21	1.80	3.36
z	0.55	1.23	1.66	3.09	5.59

Table 4.2. Distance error(%) w.r.t. rotation error added in different axes

Noise(deg)	0	0.5	1	2	4
x	2.59	2.71	4.56	4.92	7.71
y	2.59	2.73	2.98	3.01	3.36
z	2.59	2.94	3.11	3.36	4.67

Table 4.3. Distance error(%) w.r.t. added translation error

Noise(%)	0	2	5	10	15
	2.59	3.24	5.41	8.73	14.97

Baseline is another important parameter of 3D reconstruction. Three different datasets (as described at the beginning of this section) were used to test the effect of short, medium and large baselines on reconstruction precision. Fig. 4.23 shows the distance error while Fig. 4.24 shows the normal error with respect to each baseline. Of course, shorter baseline has higher error rate, which is a well known fact for stereo reconstruction. However, homography errors are smaller in case of short and medium base distances (see Fig. 4.25), hence overall reconstruction performence is better for these datasets.

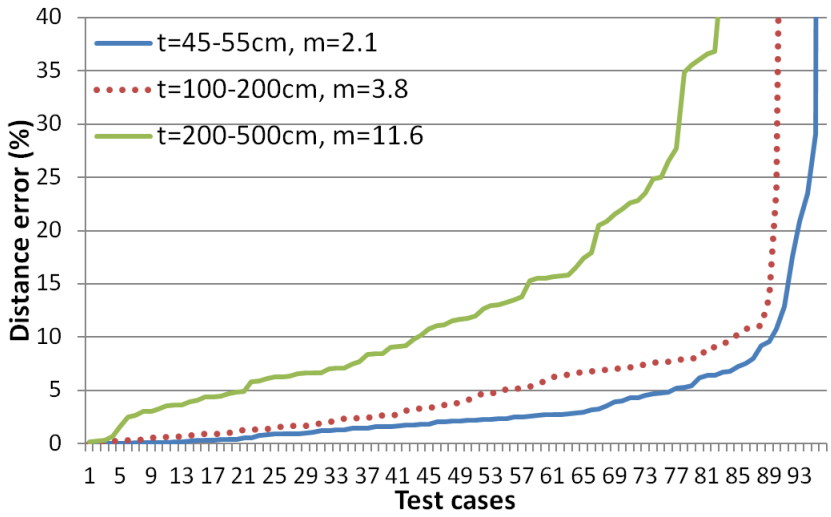

Figure 4.23. Distance error plots w.r.t. different baselines (test cases sorted independently, m is the median of errors).

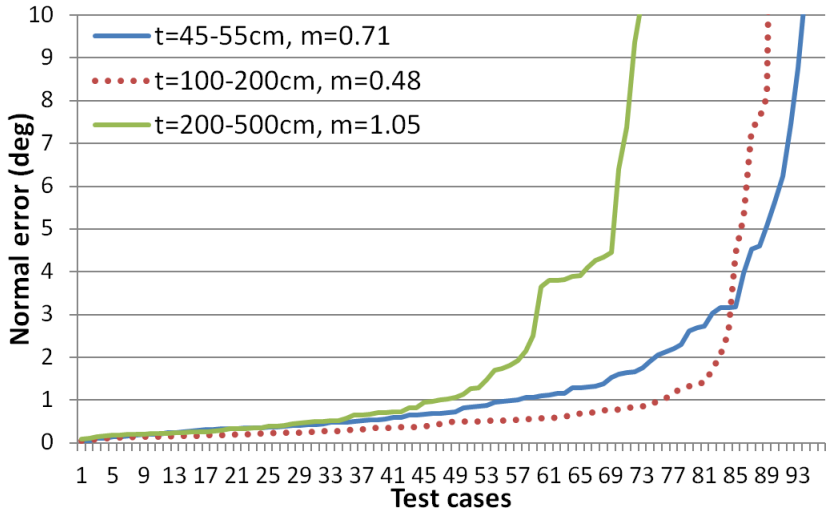

Figure 4.24. Normal error plots w.r.t. different baselines (test cases sorted independently, m is the median of errors).

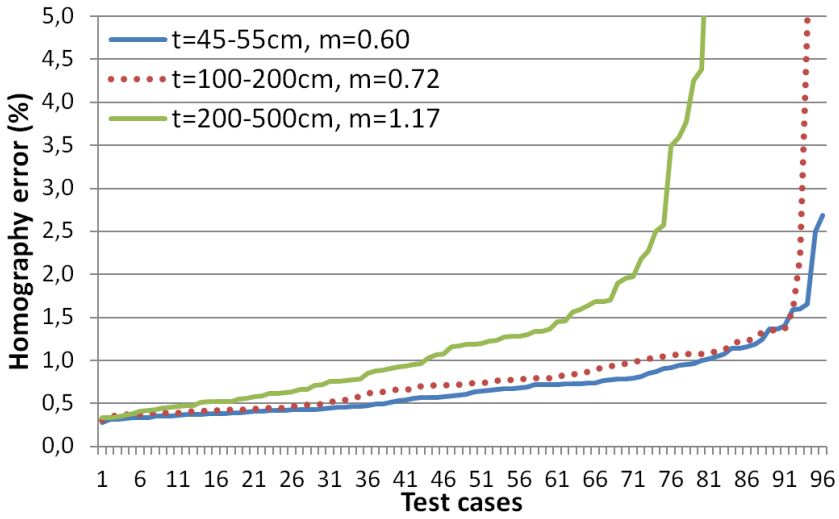

Figure 4.25. Homography error w.r.t. different baselines (test cases sorted independently, m is the median of errors).

4.5 Simultaneous Relative Pose Estimation and Plane Reconstruction

In contrast to the methods presented in the previous sections, where a plane induced homography was first estimated between image regions, then the relative pose of the cameras was factorized, finally being able to calculate the parameters of the plane based on these estimated values, here we present a simultaneous solution for all the above problems for perspective cameras.

4.5.1 Methodology

Starting from the absolute pose of perspective cameras, as described in Chapter 3.2.2, we can work directly with the normalized images (3.9)

$$\mathbf{x} = \mathbf{K}^{-1}\tilde{\mathbf{x}} \cong [\mathbf{R}|\mathbf{t}]\mathbf{X}_{\mathcal{W}}. \tag{4.33}$$

Let us formulate the relation between a given scene plane π and its images $\mathcal{D}^0$ and $\mathcal{D}^1$ in two normalized cameras (see Fig. 4.26). Assuming that the first camera coordinate system $\mathcal{C}_0$ is the reference frame, let us represent π by its unit normal $\mathbf{n} = (n_1, n_2, n_3)^\top$ and distance d to the origin. Furthermore, the relative pose of the second camera frame $\mathcal{C}_1$ is a 3D rigid body transformation $(\mathbf{R}^1, \mathbf{t}^1) : \mathcal{C}_0 \rightarrow \mathcal{C}_1$ composed of a rotation $\mathbf{R}^1$ and translation $\mathbf{t}^1$, acting between the camera frames $\mathcal{C}_0$ and $\mathcal{C}_1$. Thus the image in the first and second camera of any homogeneous 3D point $\mathbf{X}$ of the reference frame is given by

$$\mathbf{x}_{\mathcal{C}_0} \cong [\mathbf{I}|\mathbf{0}]\mathbf{X} \quad \text{and} \quad \mathbf{x}_{\mathcal{C}_1} \cong [\mathbf{R}^1|\mathbf{t}^1]\mathbf{X}. \tag{4.34}$$

The mapping of 3D plane points $\mathbf{X}_\pi \in \pi$ into the cameras $\mathcal{C}_i, i = 0, 1$ is governed by the same equations, giving rise to a planar homography $\mathbf{H}^1_\pi : \mathcal{D}^0 \rightarrow \mathcal{D}^1$ induced by $\pi = (\mathbf{n}, d)$ between the image regions $\mathcal{D}^0$ and $\mathcal{D}^1$. $\mathbf{H}^1_\pi$ is bijective (unless π is going through the

camera center, in which case π is invisible), composed up to a scale factor as

$$\mathbf{H}_{\pi}^{1} \propto \mathbf{R}^{1} - \frac{1}{d}\mathbf{t}^{1}\mathbf{n}^{\top}. \tag{4.35}$$

Thus for any point $\mathbf{X}_{\pi} \in \pi$, we have the following relation between the corresponding normalized image points $\mathbf{x}_{\mathcal{C}_0}$ and $\mathbf{x}_{\mathcal{C}_1}$:

$$\mathbf{x}_{\mathcal{C}_1} \cong \mathbf{H}_{\pi}^{1}\mathbf{x}_{\mathcal{C}_0} \cong (\mathbf{R}^{1} - \frac{1}{d}\mathbf{t}^{1}\mathbf{n}^{\top})\mathbf{x}_{\mathcal{C}_0}. \tag{4.36}$$

The classical solution is to find at least 4 such point matches and solve for $\mathbf{H}_{\pi}^{1}$, then factorize $\mathbf{R}^{1}$, $\mathbf{t}^{1}$, and $\mathbf{n}$ from $\mathbf{H}_{\pi}^{1}$ (d cannot be recovered due to the free scaling factor) [134]. However, the extraction of point correspondences in urban environment can be challenging due to repetitive structures and textureless facades, while planar regions are easier to segment and matching between frames is not affected by repetitive structures if limited camera movement is assumed. Therefore our region-based approach [Frohlich, Kato, 2018] can robustly recover the alignment of non-linear shape deformations via the solution of a special system of equations without established point correspondences. In particular, we will show that by identifying a pair of planar regions in two camera images, the relative pose as well as the 3D plane parameters can be solved up to scale without establishing any further correspondences between the regions. Of course, this is just the necessary minimal configuration. The more such regions are available, a more stable solution is obtained. Furthermore, when more cameras are available, then a special region-based bundle adjustment can be constructed within the same algebraic framework.

Following the idea of [78] we can avoid the need of working with point correspondences by integrating out both sides of (4.36), yielding the following integral equation:

$$\int_{\mathcal{D}^1} \mathbf{x}_{\mathcal{C}_1}\,\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = \int_{\mathcal{D}^0} \mathbf{H}_{\pi}^{1}\mathbf{x}_{\mathcal{C}_0}|\mathbf{J}_{\mathbf{H}_{\pi}^{1}}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}, \tag{4.37}$$

where the integral transformation $\mathbf{x}_{\mathcal{C}_1} = \mathbf{H}_{\pi}^{1}\mathbf{x}_{\mathcal{C}_0}$, $\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = |\mathbf{J}_{\mathbf{H}_{\pi}^{1}}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}$ has been applied. Since $\mathbf{H}_{\pi}^{1}$ is a 3×3 homogeneous matrix with only 8 DoF, we will set its last element to 1. Note that the above equality is true for inhomogeneous point coordinates $\mathbf{x}_{\mathcal{C}_i}$, which are obtained by projective division. The Jacobian determinant $|\mathbf{J}_{\mathbf{H}_{\pi}^{1}}| : \mathbb{R}^2 \to \mathbb{R}$ gives the measure of the transformation at each point [78].

The above equation corresponds to a system of 2 equations only, which is clearly not sufficient to solve for all parameters. As it has been previously shown in [78], applying an appropriate set of functions on both sides of an equality $a = b$ it remains valid for $f(a) = f(b)$, thus enabling us to construct new equations. Indeed, (4.36) remains valid when a function $\omega : \mathbb{R}^2 \to \mathbb{R}$ is acting on both sides of the equation, yielding the integral equation

$$\int_{\mathcal{D}^1} \omega(\mathbf{x}_{\mathcal{C}_1})\,\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = \int_{\mathcal{D}^0} \omega(\mathbf{H}_{\pi}^{1}\mathbf{x}_{\mathcal{C}_0})|\mathbf{J}_{\mathbf{H}_{\pi}^{1}}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}. \tag{4.38}$$

Adopting a set of nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, each ω_i generates a new equation yielding a system of ℓ independent equations. Hence we are able to generate sufficiently many equations. According to [78], power functions are computationally favorable, thus in our experiments, we adopted the following functions up to power o:

$$\omega_i(\mathbf{x}) = x_1^{m_i}x_2^{n_i}, \text{ with } 0 \le m_i, n_i \le o \tag{4.39}$$

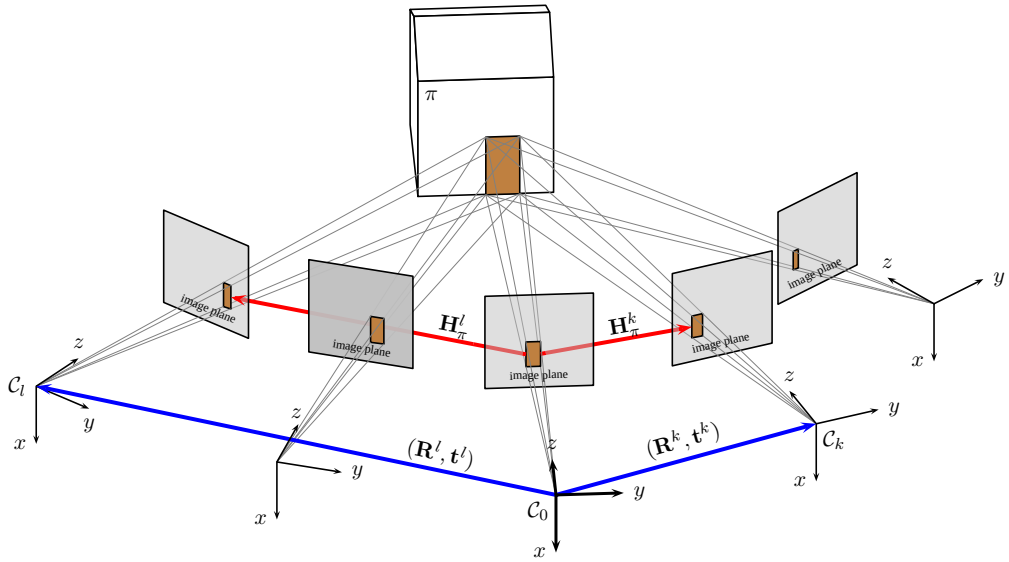

Figure 4.26. Projection of a 3D plane π in a multi-view camera system.

The unknown relative pose $(\mathbf{R}^1, \mathbf{t}^1)$ and 3D plane parameters $(\mathbf{n}, d)$ are then simply obtained as the solution of the nonlinear system of equations (4.38). $\mathbf{H}^1_\pi$ has 8 degree of freedom (DoF), because $\mathbf{n}$ is a unit vector with 2 DoF and $\mathbf{t}/d$ can only be obtained up to scale, so it has only 3 DoF. Thus we need 8 equations which can be constructed using ω_i functions from (4.39) with $0 \leq m_i, n_i \leq 2$ and $m_i + n_i \leq 3$. In practice, however, an overdetermined system is constructed, which is then solved in the *least squares sense* by minimizing the algebraic error via a standard *Levenberg-Marquardt* algorithm.

Reconstruction of Multiple Regions

Let us now investigate the case, when a pair of cameras is observing multiple 3D scene planes. Each plane π_i generates a homography $\mathbf{H}^1_{\pi_i}$ between the corresponding image regions $\mathcal{D}^0_i$ and $\mathcal{D}^1_i$. While (4.36) and (4.38) remain valid for each of these homographies, note that the relative pose $(\mathbf{R}^1, \mathbf{t}^1)$ of the cameras is the same for all $\mathbf{H}^1_{\pi_i}$, they only differ in the 3D plane parameters $(\mathbf{n}_i, d_i)$. Hence for all $\{\pi_i\}_{i=1}^N$, we have

$$\mathbf{x}_{\mathcal{C}_1} \cong \mathbf{H}^1_{\pi_i}\mathbf{x}_{\mathcal{C}_0} \cong (\mathbf{R}^1 - \frac{1}{d_i}\mathbf{t}^1\mathbf{n}_i^\top)\mathbf{x}_{\mathcal{C}_0}, \text{ with } \mathbf{x}_{\mathcal{C}_0} \in \mathcal{D}^0_i \text{ and } \mathbf{x}_{\mathcal{C}_1} \in \mathcal{D}^1_i \tag{4.40}$$

and (4.38) becomes a system of N equations [Frohlich, Kato, 2018] in terms of the common camera pose $(\mathbf{R}^1, \mathbf{t}^1)$ and the parameters $(\mathbf{n}_i, d_i)$ of the 3D planes $\{\pi_i\}_{i=1}^N$:

$$\int_{\mathcal{D}^1_i} \omega(\mathbf{x}_{\mathcal{C}_1})\,\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = \int_{\mathcal{D}^0_i} \omega(\mathbf{H}^1_{\pi_i}\mathbf{x}_{\mathcal{C}_0})|\mathbf{J}_{\mathbf{H}^1_{\pi_i}}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}, \quad 1 \leq i \leq N \tag{4.41}$$

For a given ω function, the above equations provide N constraints on the relative pose parameters, but only 1 constraint for each plane π_i, having a total of N equations. Note also, that we have one free scaling factor for the whole system in (4.41), because a relative d_i parameter for the planes need to be determined, only one of them can be set freely. Therefore the minimal number of equations needed to solve for 2 cameras and $N \geq 1$ planes is $E = 6 + 3N - 1$. In terms of the necessary powers of ω_i functions in (4.39), o should satisfy $1 + o(o+2) \geq E$.

Multi-view Reconstruction

When multiple cameras are observing the scene planes, then we can construct a system of equations which contains multiple constraints not only for the camera relative poses but also for each 3D plane. This way, we obtain a region-based bundle adjustment, where all camera pose parameters and all 3D plane parameters are simultaneously solved [Frohlich, Kato, 2018]. Let us have a closer look at these equations. First of all, a reference camera frame $\mathcal{C}_0$ is chosen, which provides the reference coordinate system of the whole camera system: each camera's relative pose is determined w.r.t. $\mathcal{C}_0$ and all planes are reconstructed within $\mathcal{C}_0$. Assuming that all scene planes $\{\pi_i\}_{i=1}^{N}$ are visible in every camera $\{\mathcal{C}_k\}_{k=0}^{M-1}$, each plane π_i generates a homography $\mathbf{H}_{\pi_i}^{k}$ between the corresponding image regions in the reference camera $\mathcal{D}_i^0$ and the k^{th} camera $\mathcal{D}_i^k$:

$$\forall 1 \leq k \leq M-1: \quad \mathbf{x}_{\mathcal{C}_k} \cong \mathbf{H}_{\pi_i}^{k}\mathbf{x}_{\mathcal{C}_0} \cong (\mathbf{R}^k - \frac{1}{d_i}\mathbf{t}^k\mathbf{n}_i^\top)\mathbf{x}_{\mathcal{C}_0}. \tag{4.42}$$

Hence each camera provides a new constraint on the scene plane parameters $(\mathbf{n}_i, d_i)$, yielding a total of $M-1$ constraints for reconstructing π_i. If a particular plane is not visible in all other cameras, then the number of these constraints is reduced. As long as a particular plane π_i is visible in the reference camera and at least one other camera k, then it is possible to reconstruct it using the equations constructed from the above homography, just like in the minimal case discussed in Chapter 4.5.1.

A particular camera pair $(\mathcal{C}_0, \mathcal{C}_k)$ provides N equations in terms of the common camera pose $(\mathbf{R}^k, \mathbf{t}^k)$ and the parameters $(\mathbf{n}_i, d_i)$ of the 3D planes $\{\pi_i\}_{i=1}^{N}$, yielding a system of N equations similar to (4.38). Therefore we get

$$\int_{\mathcal{D}_i^k} \omega(\mathbf{x}_{\mathcal{C}_k})\,\mathrm{d}\mathbf{x}_{\mathcal{C}_k} = \int_{\mathcal{D}_i^0} \omega(\mathbf{H}_{\pi_i}^{k}\mathbf{x}_{\mathcal{C}_0})|\mathbf{J}_{\mathbf{H}_{\pi_i}^{k}}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}, \quad 1 \leq i \leq N \text{ and } 1 \leq k \leq M-1 \tag{4.43}$$

For a given ω function, the above equations provide N constraints on each relative pose $(\mathbf{R}^k, \mathbf{t}^k)$, and $M-1$ constraints for each plane π_i, having a total of $N(M-1)$ equations. The minimal number of equations needed to solve for $M \geq 2$ cameras and $N \geq 1$ planes is $E = 6(M-1) + 3N - 1$. In terms of the necessary powers of ω_i functions in (4.39), o should satisfy $1 + o(o+2) \geq E$.

Algorithmic Solution

The algorithmic summary of the proposed method for an arbitrary $(M \geq 3)$ multi-view camera system is presented in Algorithm 5. The first part of the algorithm is to solve for each neighboring camera pair, that will provide the initial parameters for the second part. This step does not require any specific initialization of the parameters, except that $d_i = 0$ should be avoided. Since plane distance is expressed as the distance from the plane to the origin along the surface normal vector's direction, it is a positive number if the origin is on the same side of the plane as the normal, thus it can be initialized with an arbitrary positive value, in our tests we used the initialization $d_i = 7$. Since plane normal is of unit length, it has only 2 DoF, the third parameter is always calculated with the criteria that the normal should point towards the camera. Since each pairwise solution provides a reconstruction in one of the cameras, these have to be transformed into the common reference frame of $\mathcal{C}_0$,

Algorithm 6 The proposed multi-view simultaneous algorithm

Input: $M \geq 3$ 2D image masks with $N \geq 1$ corresponding planar regions
Output: Relative pose of the cameras w.r.t. $\mathcal{C}_0$, reconstruction $(\mathbf{n}_i, d_i)$ of the N planes

1: **Pairwise step:** For each $M-1$ neighboring camera pair:
2: Initialize pose parameters with $[\mathbf{I}|\mathbf{0}]$ and plane parameters with $n = (0,0,-1)^T$, $d = 7$
3: Construct $\mathbf{H}_\pi$ using (4.40) and divide it by its last element
4: Construct and solve the system of equations of (4.41)
5: **Multi-view step**: Choose reference camera $\mathcal{C}_0$
6: Write up relative poses w.r.t. $\mathcal{C}_0$ as (4.45) and transform reconstruction $(\mathbf{n}_i, d_i)$ parameters into $\mathcal{C}_0$ reference frame using (4.44)
7: Initialize reconstruction based on the filtered camera pairs
8: Write up $\mathbf{H}^k_{\pi_i}$ (devided by its last element) for each camera pair $(\mathcal{C}_0, \mathcal{C}_k)$ using (4.42)
9: Construct and solve the system of equations of (4.43) for M cameras and N planes simultaneously

that is practically chosen the middle camera. Plane parameters $(\mathbf{n}^0_i, d^0_i)$ are obtained from $(\mathbf{n}^k_i, d^k_i)$ as

$$\mathbf{n_i}^0 = \mathbf{R}^{kT}\mathbf{n}^k_i \quad \text{and} \quad d^0_i = d^k_i + (\mathbf{n}^k_i)^T\mathbf{t}^k \tag{4.44}$$

Relative poses also have to be expressed in the $\mathcal{C}_0$ reference frame. For any camera $\mathcal{C}_l$ that has a relative pose $(\mathbf{R}^{k,l}, \mathbf{t}^{k,l})$ defined to its neighbor C_k, whose relative pose $(\mathbf{R}^{0,k}, \mathbf{t}^{0,k})$ w.r.t. $\mathcal{C}_0$ is already known, then the relative pose of C_l in the reference frame will be

$$(\mathbf{R}^{0,l}|\mathbf{t}^{0,l}) = (\mathbf{R}^{k,l}|\mathbf{t}^{k,l})(\mathbf{R}^{0,k}|\mathbf{t}^{0,k}), \tag{4.45}$$

where $(\mathbf{R}|\mathbf{t})$ denote the homogeneous 4×4 matrix constructed from $\mathbf{R}$ and $\mathbf{t}$.

Since multiple camera pairs will provide alternative initializations for the reconstruction, and some pairs might be solved less precisely than others, we have to filter the camera pairs. Comparing the algebraic error of the pairwise reconstructions we filter out the pair with the highest error if it's above the experimentally determined threshold of $5e^{-9}$, and if it's bigger than 3 times the median errors of the camera pairs. The reconstruction parameters of the remaining pairs are simply averaged out and together with the relative poses expressed w.r.t. $\mathcal{C}_0$ (4.45) provide the input for the multi-view step.

For the numerical implementation of the equations we also included the alternative forms of the equation using the inverse transformation and the reverse integral transformation as described in [78]. These mathematically redundant forms don't provide extra constraints for the parameters, but increase the numerical stability of the method.

4.5.2 Experimental Synthetic Results

For the quantitative evaluation of the proposed approach, a benchmark dataset is generated with a greatly simplified real world urban environment in mind. The synthetic data is not metric, but we can interpret it as having the planar shapes in the 3D scene represent 1×1 m regions, which scales everything into a metric interpretation for easier understanding. A scene was created by placing 3 different planar shapes in 3D space having $\pm 20^\circ$ relative rotation around the vertical or horizontal axis and translated by $1-2$ m in the horizontal and vertical direction, while $1-3$ m in depth. The scene is then captured by a 1Mpx virtual camera placed at the initial distance of 4 m from the middle plane, then moved into

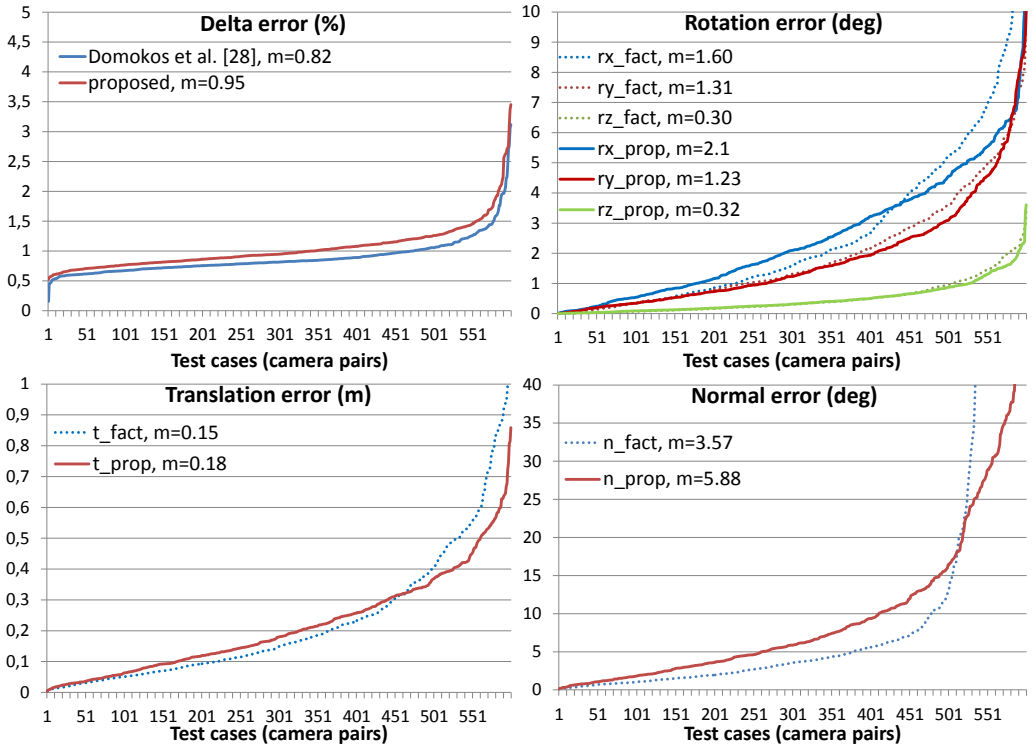

Figure 4.27. Comparison to homography estimation method of [78] in terms of δ error of the projections, and relative camera pose and plane parameters (factorized from $\mathbf{H}$ by [78], provided directly by proposed). (m stands for median).

5 different random positions, in such a way that the resulting positions form a movement trajectory (see Fig. 4.26). Between each frame there is a random relative movement of up to 0.5 m and a rotation of $\pm 5^\circ$ around the vertical axis, and $\pm 2^\circ$ around the other two axes. The binary images captured by the camera are the input parameters of our algorithm, having the correspondence between the regions provided. Practically there is no limitation on the size difference of the projections. The results were quantitatively evaluated in the pose and reconstruction parameters, as well as in terms of the aligning homography, which is characterized by the percentage of the non-overlapping areas between the aligned regions (denoted as δ error).

Minimal Case

The minimal case consists of one plane seen by two cameras, where we have to estimate a single homography aligning the planar image regions. First of all, we compared our method to the homography estimation method of [78] that solves a similar system of equations but it is parametrized in terms of 8 elements of $\mathbf{H}_\pi^1$, while our method uses the $\mathbf{R}^1$, $\mathbf{t}^1$, $\mathbf{n}$, d parametrization. In Fig. 4.27, the first plot shows that despite having a different parametrization, the stability of the proposed method remains similar. All synthetic plots are sorted based on the error values. We have to highlight here that while the first method only estimates a homography matrix with 8 DoF, the proposed method estimates the parameters of the relative pose $\mathbf{R}^1$, $\mathbf{t}^1$, the 3D plane reconstruction $\mathbf{n}$, d, as well as the composed aligning homography $\mathbf{H}_\pi^1$ simultaneously, up to a scale factor.

Of course, we can decompose the homography matrix computed by the method of [78] in terms of $(\mathbf{R}, \mathbf{t})$ and $(\mathbf{n}, d)$ using the standard decomposition of [134] which uses the

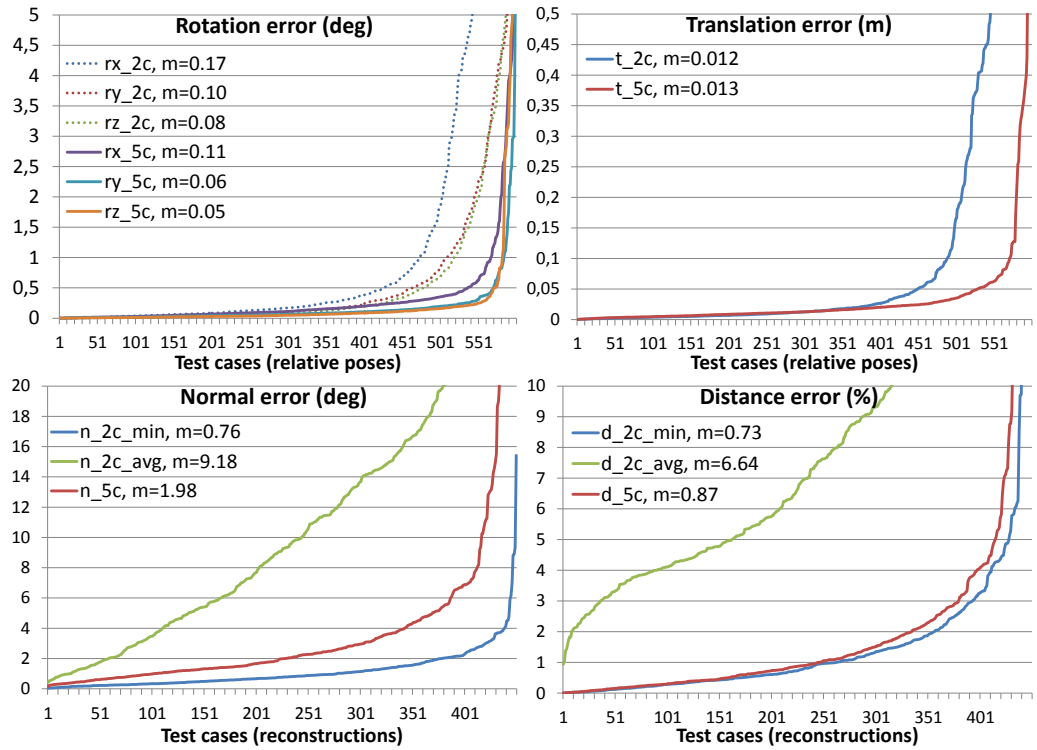

Figure 4.28. Bundle adjustment refinement over 5 cameras compared to a pairwise solution, both with 3 regions (m stands for median).

singular value decomposition (SVD) of the homography. In Fig. 4.27, we compare these decomposed parameters with our results. We can observe that while the pose parameters are obtained with similar precision, the plane reconstruction has slightly higher error in median, but also shows increased robustness in the last part of the plots. These results match well with the slightly higher δ errors shown on the first plot in Fig. 4.27. We should highlight here, that while this was a fair comparison to the baseline method [78], our method's advantage is the ability to handle multiple regions and camera images, since it will provide an optimized solution for the camera group with a common scale, instead of having independent solutions for each region and camera pair.

Multi-View Reconstruction

As it was shown in Chapter 4.5.1, multiple cameras theoretically provide more constraints for the reconstruction, while more planes on the pose parameters. To confirm this, we evaluated the proposed method with 5 cameras in two different setups: First solving a pairwise reconstruction for each neighboring camera pair, then in the second setup solving for all 5 cameras, using the full algorithm as presented in Algorithm 6. In both cases, 3 planes were used and the results were compared to the synthetic reference values, that were also used to correctly scale the translation and plane distance parameters that are estimated only up to a scale factor. The relative pose parameters are evaluated as absolute errors in the rotation angles, and the difference in the total translation (see first row of Fig. 4.28). We can observe, that the relevant improvement of the multi-camera setup is not necessarily visible in the median error levels, but more so in the number of correct solutions. The number of camera pairs solved with a relative pose error lower than 0.5° in rotation and 5 cm in translation is increased from 75% to above 90%.

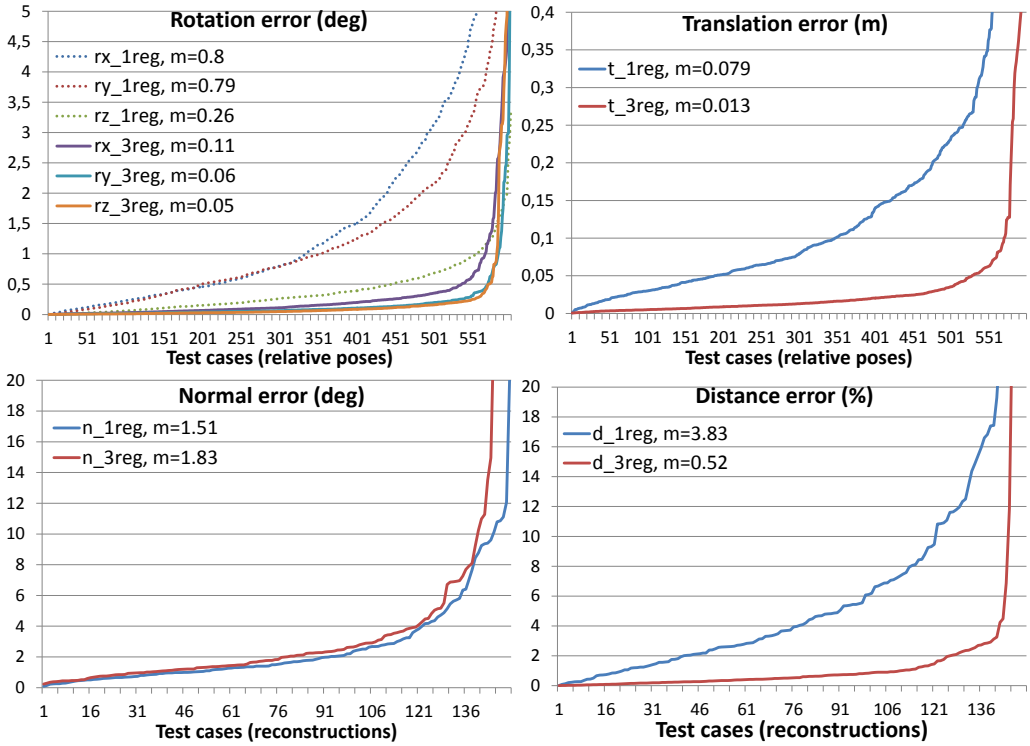

Figure 4.29. 5 camera results with 1 region compared to 3 regions evaluated in terms of the pose errors (first row) and reconstruction parameters (second row), that were evaluated on the same regions for the 1 and 3 region test cases.

Since the 4 camera pairs in each test case were used separately in the 2-camera pairwise setup, the reconstruction they provide will be 4 different ones for the same plane, thus in the second row of Fig. 4.28 we show both the minimum and average $(\mathbf{n}, d)$ errors of these for each test case. In contrast, the bundle adjustment multi-view setup provides one single reconstruction in the reference camera frame. We can see, that the bundle adjustment step greatly reduces the mean errors that the pairwise solution had, approaching to the minimal errors. Note that d is evaluated as the difference of the scaled result and the reference distance, expressed as a percentage of the ground truth.

Single Plane Reconstruction

An interesting scenario is a multi-view setup with only one plane available, therefore the method was evaluated for the $M = 5$ and $N = 1$ setup. Results were compared to those obtained on the full dataset using 3 planar regions in each test case, to evaluate the improvements given by the higher number of planes. As can be seen in the first row of Fig. 4.29, the rotation and translation parameters of the relative pose are greatly improved due to multiple different planes. In more than 90% of the cases, all rotation errors were well below 0.5° in the 3 region setup, while in the single region case the errors are below 1° in half of the test cases only. The translation parameters show the same improvement: from a median error of 8 cm reducing errors to below 5 cm in 88% of the cases.

The reconstruction error plots in the second row of Fig. 4.29 show the errors in the normal vector angle, and the plane distance. While in the first row on the horizontal axis, the number of camera pairs were depicted which (having a reference dataset of 150 test cases each with 5 consecutive frames of a scene with 3 regions) consists of a total of 600

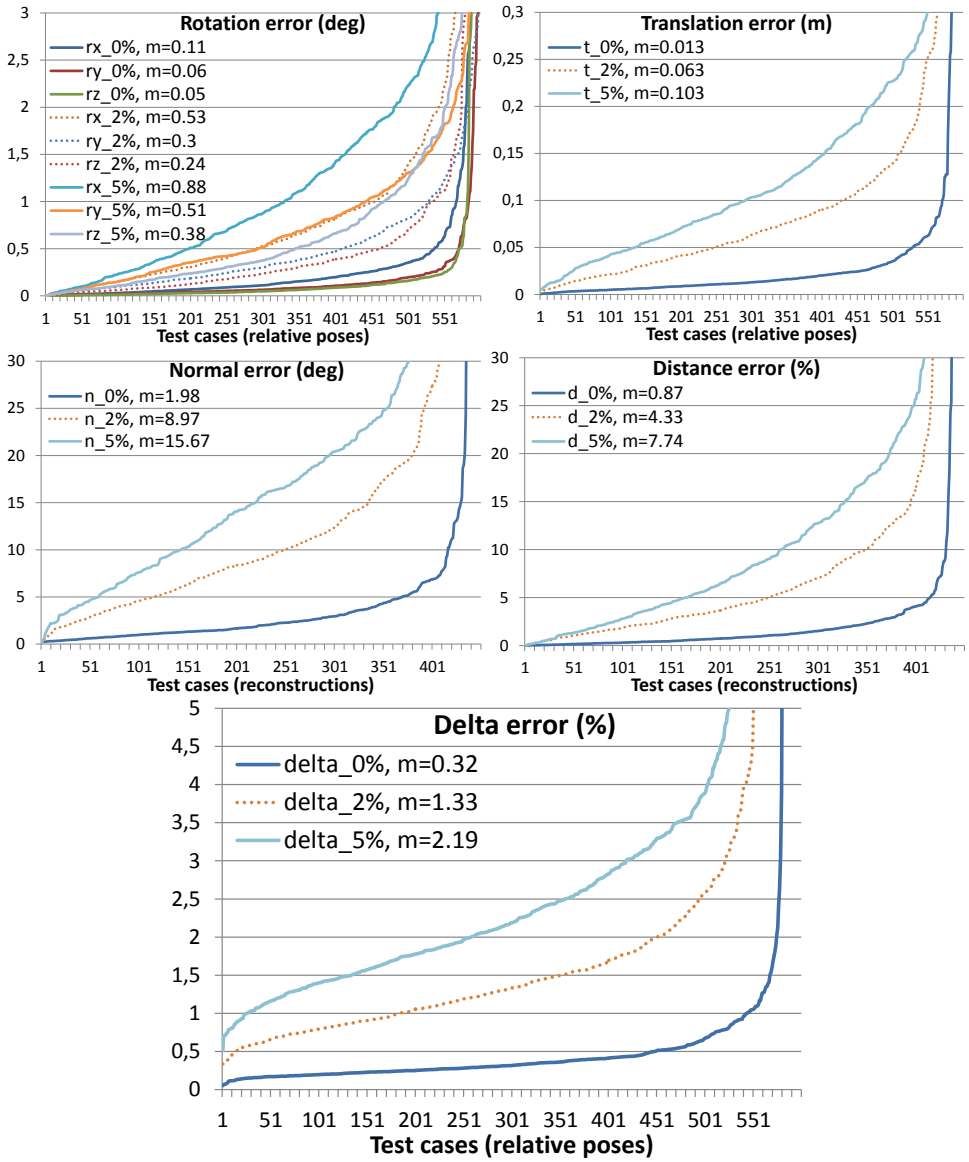

Figure 4.30. Segmentation error test results in terms of pose and reconstruction parameter errors both in the case of perfect regions and regions with simulated segmentation errors of 2% and 5%. δ error measured on the perfect regions.

relative poses; in case of the reconstruction parameters we only have to refer to the separate test cases since in each test case multiple cameras estimate one common reconstruction of the scene. According to the plots in Fig. 4.29, we can observe that the plane normal is less affected by the reduced number of planes, while the plane distance parameter is affected in a quantity comparable to the pose parameters, which is expected, since up to a point these parameters are able to compensate each other if not enough constraints are given. But using only two extra regions, the results can drastically improve: in 85% of the test cases, the distance of the plane is estimated with a relative error less then 2% instead of 10%.

Robustness Against Segmentation Errors

In order to evaluate the robustness of the proposed algorithm against the inevitable errors occuring in image segmentation, we simulated segmentation errors by randomly changing pixels around the contour of the regions by an amount of 2% and 5% of the size of the

region. Herein, all tests were run in the $M = 5$ cameras and $N = 3$ planes setup.

The errors in the estimated relative pose on these specific datasets can be seen in the first row of Fig. 4.30, where first the rotation errors of all the relative poses are plotted, grouped by the axes, then the translation errors are shown next. All plots are sorted in a best to worst sense by each parameter separately, and the results are compared to the base dataset which uses perfect segmentations. In the second row of Fig. 4.30, the reconstruction errors are shown, that were estimated simultaneously with the relative pose by the proposed algorithm. Both plane normal and distance errors are evaluated in light of the segmentation errors.

Analyzing the pose and reconstruction parameters at the same time, one can observe that the segmentation errors have a similar impact on all the parameters, but still the median rotation errors don't exceed 0.5° except the rotation around the vertical X axis in the 5% segmentation error case. Based on the relative pose results, 5% segmentation error could be acceptable in many applications, where 10 cm translation errors are acceptable, but due to the reconstruction being more sensitive to these, a segmentation error of less than 2% would be desirable in most applications. On the last plot in Fig. 4.30, the δ errors are shown. We can see that in about 66% of the cases, a δ error of less than 3% is achieved even in the presence of segmentation error. Based on our previous experiences, in many applications a δ error of up to 5% is considered a correct solution.

4.5.3 Real Data Experiments

In the first real data test case, we present the results on a high resolution 2D-3D dataset, that contains ground truth pointcloud data captured by a precise Lidar scanner, 4K resolution UAV video frames and also the reference 3D positions of special markers placed in the scene, which enabled the calculation of reference camera poses with UPnP [30] for each camera frame, resulting a median forward projection error of the markers of only $1 - 2$ cm. In urban environments the automatic segmentation of planar structures, windows, doors, facades or boards could be solved with different methods [135], but in our tests the segmentation of the corresponding planar regions on each frame was performed in a semi-automatic way using region growing segmentation method available in commercial image editing programs, requiring only a few clicks of user intervention. The segmented 2 regions are shown in Fig. 4.31, marked with red on the first and last image frame. We used 5 frames of the video sequence, at $1 - 2$ seconds distance from each other. The estimated parameters were compared to the ground truth values (plane parameters were calculated from the point cloud data). The relative camera pose rotations were estimated with a mean error of 0.72°, 0.2°, 0.59° around the X, Y, Z axes, the maximum rotation errors being below 1°. The relative translation was evaluated as the difference of the reference value and the correctly scaled up estimated translation, that can be interpreted as a position displacement in the metric space. These errors are between 12 cm and 33 cm. The error in the orientation of the estimated plane normals was 2° and 2.95° respectively, while the error of the plane distance from the origin was 0.38 m and 0.77 m. For a different perspective over the plane distance parameter, we also calculated the distance from the camera to the center of the reference 3D region and the reconstructed 3D region, since this might be more useful in many applications. At camera-to-surface distances of 14.1 m and 21.4 m these errors represent 3% and 7% differences, respectively. These results comply with the synthetic test results shown in Fig. 4.30, where we found that with higher segmentation error the plane distance error can go above

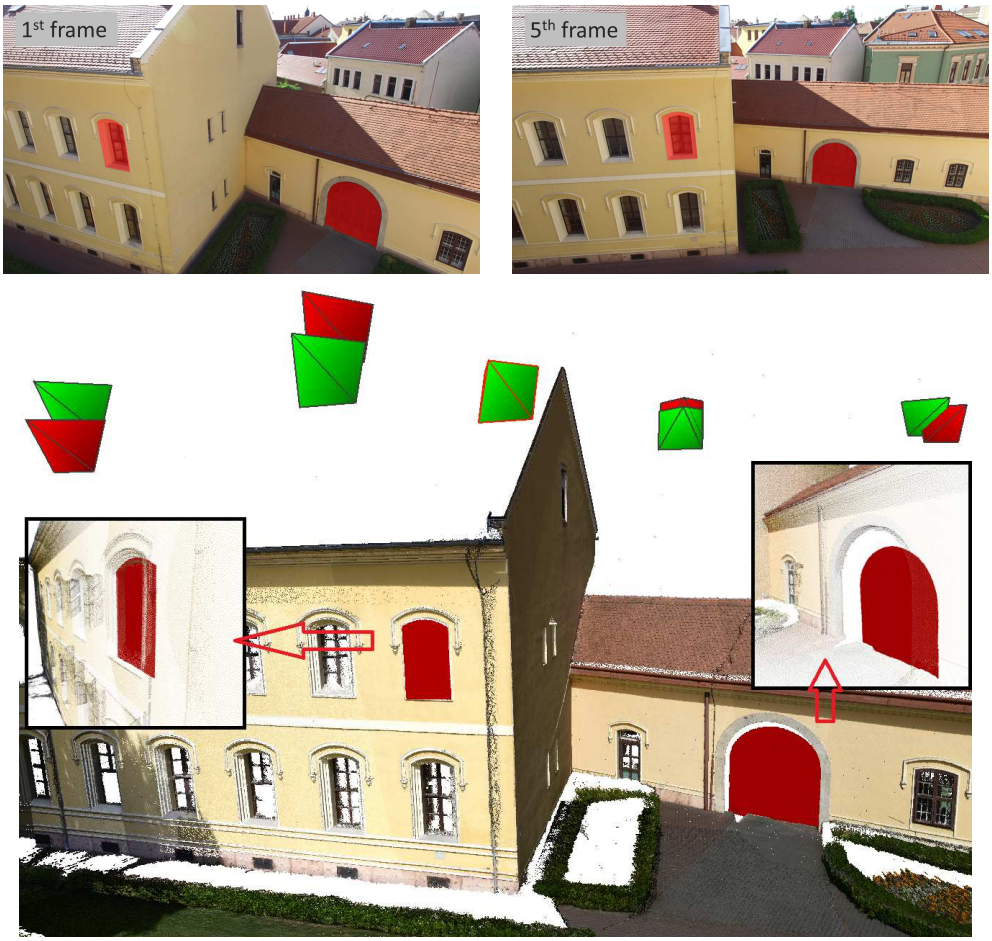

Figure 4.31. Top: first and last image of the sequence, with the 2 segmented corresponding regions marked in red. Bottom: the reference camera positions (green), the estimated camera positions (red) and the reconstructed 3D planar regions are shown (also including a side view of both).

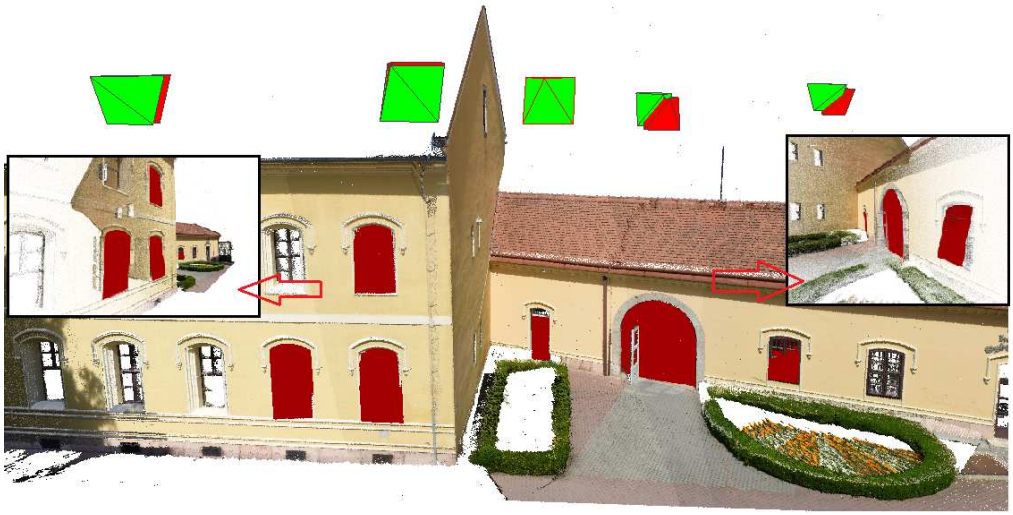

Figure 4.32. Results on the same sequence as in Fig. 4.31 but with 6 regions. The reference camera positions (green), the estimated camera positions (red) and the reconstructed 3D planar regions are shown (including close-up views from the side). Note the improvement in the relative camera poses!

7%. Increasing the number of regions by using all the segmentable regions in the same sequence (see Fig. 4.32) the mean pose errors were reduced to $0.12°$, $0.18°$, $0.13°$ rotation and $2.7 - 16$ cm translation; and the median reconstruction errors were $5°$ and 65 cm, thus we considered test cases with less regions are more interesting to show, since more regions obviously increase the stability of the algorithm.

Comparison on KITTI Dataset

To test the applicability of the proposed method to urban road-scene applications we used the KITTI [142] dataset. Having a camera attached to a car that is only capable of forward motion and turns is a more challenging problem for pose estimation and reconstruction. A region based plane reconstruction method could be applied in such an environment for example for the reconstruction of traffic signs as planar objects. Unfortunately in the KITTI dataset not all traffic signs are visible in the 3D pointclouds, due to their different height and the position and orientation of the Velodyne scanner. We found that from higher distances these traffic signs were visible for the 3D scanner, thus we combined a sequence of pointclouds using the ground truth poses and segmented traffic signs on these combined pointclouds. Then selecting all the cases where this 3D segmentation was succesful, we had from the KITTI Visual Odometry training video sequences 41 different test cases. In each test case the segmented traffic signs on 5 consecutive frames were extracted using the tool described in the previous section, while automatic segmentation of these objects is also a well researched topic with many solutions, *e.g.* [131] also provides the boundary of the signs.

The proposed method was tested both on the minimum number of 3 frames, and on 5 frames per test case (using only one small segmented region from each frame) where the latter showed slightly better reconstruction results, the individual relative poses showed similar median errors but more robustness for the 3 camera setup. This is caused by the traffic signs moving out of the frame too fast, thus the more frames we try to use, the more segmentation error we have on the extra images. Median errors on the dataset using 5 frames were of $0.12°, 0.24°, 0.098°$ and 0.088 m in the relative poses, and $10.14°$ with 0.55 m the normal vector error and object distance of the reconstructions on the 41 testcases. Nevertheless, 80% of the cases were solved with reconstruction errors below $20°$ in normal vector, and 1 m in the object center's distance with 5 frames.

Evaluating our results in a similar way as the official KITTI Visual Odometry benchmark, only on the above described test cases with reconstruction errors below $20°$ and 1 m, using 5 frames, we get a median translation error of 5.28% and rotation error of $0.2126(deg/m)$, that is comparable to the published benchmark results of State-of-the-Art methods (*e.g.* VISO2-M [143] is only better in rotation [T$= 11.94\%$, R$=0.0234(deg/m)$], but it cannot reconstruct traffic signs).

A direct comparison with feature based multi-view reconstruction methods can only be performed using the full images as inputs instead of just the segmented traffic signs, since those typically wouldn't provide enough image features. For this we used the State-of-the-Art Structure from Motion and multi-view reconstruction library COLMAP [144, 145] that was recently rated the best of the tested 15 reconstruction methods by [146]. We used the C++ implementation with CUDA and reconstructed each test case from the same 5 frames. The median errors of the estimated camera poses and the reconstructed plane parameters are given in Table 4.4. Note that COLMAP fails to find good initial image pairs in 13 cases,

Table 4.4. Quantitative results of the proposed method and COLMAP [144, 145] on the KITTI dataset, evaluated only on test cases where reconstruction was inside the reference bounding box

	solved	inBB	norm.(°)	obj_d(m)	R(°/m)	T(%)	time
COLMAP	28/41	11/28	11.07	**0.46**	**0.13**	**2.35**	46
Proposed	**41/41**	**34/41**	**8.63**	0.47	0.17	4.53	**15**

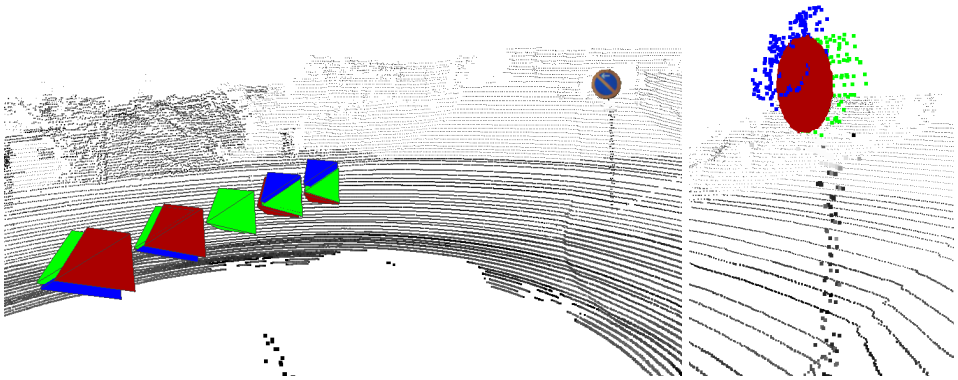

Figure 4.33. Comparative results of the propsed method and COLMAP[144, 145]: Camera poses (left) and a traffic sign reconstruction (right) shown in green (ground truth), red (proposed), and blue (COLMAP).

thus providing no solution at all, while less then half of the solved cases provided a correct reconstruction of the traffic sign that can be identified inside the 1 m bounding box of the reference. For a fair comparison we only evaluated the test cases where the traffic sign was reconstructed inside the bounding box. The proposed algorithm not only runs faster with a native Matlab implementation, but it solves all test cases and reconstructs inside the bounding box in 34 test cases. An example result of frame 220 of Sequence 03 is shown in Fig. 4.33.

4.6 Summary

In this chapter, a new homography estimation method has been proposed for central omnidirectional cameras. Unlike traditional homography estimation approaches, we work with segmented regions corresponding to a 3D planar patch, hence our algorithm avoids the need for keypoint detection and descriptor extraction. In addition, being a purely shape-based approach, our method works with multimodal sensors as long as corresponding regions can be segmented in the different modalities. The parameters of the homography are directly obtained as the solution of a system of non-linear equations, whose size is independent of the input images. Furthermore, the method is also independent of the internal projection model of the camera as long as the projection function and its gradient are known. The algorithm is computationally efficient, allowing near-real time execution with a further optimized implementation. We have presented different applications for the use of such estimated planar homographies, first, for relative pose factorization assuming some real world scene constraints and the availability of the camera's vertical direction, then for the reconstruction of the planar region. These being closed form solutions, they run in real-time which can be particularly useful for mobile and embedded vision systems. We have also proposed

a novel simultaneous reconstruction, relative pose and homography estimation method for perspective cameras. It constructs a system of non-linear equations, whose solution directly provides the relative poses of the cameras, the reconstruction of the 3D planes, as well as the aligning planar homographies between the image regions. It has been shown that with more than two cameras, a special region-based bundle adjustment provides robust results in a multi-view camera system. Quantitative evaluation on various synthetic datasets confirms the performance and robustness of the methods. We have also demonstrated, that the accuracy of our homography estimates allows reliable estimation of extrinsic camera parameters and reconstruction of planar region of superior performance w.r.t. a classical plane reconstruction algorithm. The simultaneous multi-view method was extensively validated and compared with recent methods on the KITTI dataset, where it proved State-of-the-Art performance.

Chapter 5

Conclusions

This thesis work presented the author's research on three important computer vision topics, namely pose estimation, 3D reconstruction and planar homographies. Since most of the current solutions rely on the extraction and matching of point-wise features, the unusual region-based registration formulation of the presented methods brings a novel approach to these problems. Since large field of view omnidirectional cameras are getting preferential in many modern applications, such as robotics, navigation or autonomous driving, most of the region-based methods were presented using a general spherical camera model that is valid for several types of dioptric and catadioptric cameras, and can also be applied for the special case of traditional perspective cameras. The presented work intended to provide alternative solutions for well researched computer vision problems. The presented region-based absolute pose estimation brings multiple advantages. Besides the fact that the use of point features is completely avoided, in many real world scenes with homogeneous untextured surfaces large smooth surfaces might be easier to identify than robust point features, since such corresponding regions are easily detectable across different modalities, even in 3D without intensity information. This holds a large potential in many applications in fields where multispectral, hyperspectral or IR imaging is used in combination with depth data, such as cultural heritage, for which the thesis proposed two new 2D-3D data fusion methods. Homography estimation is also mainly solved using well selected corresponding points, thus the presented region-based homography estimation brings an alternative solution with the advantages mentioned above. Since many applications rely on a segmentation of the input images, such planar regions may already be segmented out in a processing pipeline (*e.g.* industrial production line surveillance, urban traffic signs or building facade detection), thus the homography estimation can be straightforward, not needing any manual user input, or extra feature detection that could fail due to the non-linear distortion of omnidirectional cameras. Based on the well known relation between the homography, relative pose and inducing plane parameters [15], multiple solutions are presented for estimating the relative pose between cameras, and reconstruction of the plane, both by factorization from an estimated general homography, and by parameterizing the homography estimation problem itself through these parameters to have a direct solution of the problem.

The first topic of the thesis addressed a novel registration framework for the absolute pose estimation of a camera with respect to a reference 3D coordinate frame, without using explicit point correspondences. The solution relies solely on segmented 2D-3D planar patches. As little as one such segmented region pair is enough to estimate the extrinsic parameters of the camera, but more regions increase the robustness and precision of the

method. The proposed method is general enough to be used both for perspective and omnidirectional central cameras. The method was validated on large synthetic datasets, and on various real data test cases. Two applications were proposed focusing on cultural heritage, first a pose estimation based on the extension of the method to non-planar regions, then a 2D-3D visual data fusion method, that described a full pipeline in which the proposed pose estimation step can be included, but focused more on the problem of visual data fusion and correct camera selection in case of large number of camera images. The pipeline was tested on the large scale dataset of two Reformed churches.

The second topic addressed a region-based homography estimation method valid for central cameras, that works with segmented regions corresponding to the same 3D planar patch, hence it avoids the use of keypoints. Being a purely shape-based approach, the method works with multimodal sensors as long as corresponding regions can be segmented in the different modalities. The method is computationally efficient, and independent of the internal projection model of the camera as long as the projection function and its gradient are known. Two applications based on closed form solutions were proposed that rely on planar homographies estimated this way, one for the factorization of the cameras' relative pose, another for the reconstruction of the planar region. Quantitative evaluation on various synthetic datasets confirms the performance and robustness of the methods, reconstruction of planar regions showed superior performance w.r.t. a classical plane reconstruction algorithm. A novel simultaneous reconstruction, relative pose and homography estimation method was also proposed, that relies on the construction of a system of non-linear equations, whose solution directly provides the relative pose of the cameras, the 3D planar reconstruction of the region, as well as the aligning planar homographies between the image regions. This method also relies on the 2D segmentation of planar regions, but a special region-based bundle adjustment setup is applied, thus it can handle more than two cameras in an optimal way. Validated on the KITTI dataset, the method proved State-of-the-Art performance.

Appendix A

Summary in English

Computer vision is the scientific field that aims at analyzing and interpreting digital images to gain higher-level understanding through the use of various computational tools. One of the fundamental tasks is to determine the position and orientation of a camera in the world, *i.e.* estimate its absolute pose relative to a reference coordinate frame. Having at least two images with known pose in a common coordinate frame directly enables us to reconstruct the missing depth information of the scene, practically in the same way as the human visual system does. The pose estimation is a vital step of any computer vision algorithm, while 3D reconstruction is also often needed in real world applications. Since our goal was to propose novel region-based solutions for different problems, we could also make use of planar homographies to gain a different approach of the problems. This work presents my research on developing solutions for various problems related to pose estimation and 3D reconstruction.

A.1 Key Points of the Thesis

In the following, I summarized my results into two main thesis groups. In the first one, I present my findings on 2D-3D absolute pose estimation and visual data fusion, while in the second one my results on planar homography estimation and 3D reconstruction are shown. In Table A.1., the connections between the thesis points and the corresponding publications are displayed.

I. **Absolute Pose Estimation and Data Fusion**
Inspired by the 2D registration framework of [78], [77] proposed a novel formulation of the absolute pose estimation of a perspective camera with respect to a 3D depth data as a general 2D-3D registration that works without the use of any dedicated calibration pattern or explicit point correspondences. This idea can be extended into a general framework for the absolute pose estimation of central spherical cameras, and applied for different visual data fusion tasks. The basic idea is to set up a system of non-linear equations whose solution directly provides the parameters of the aligning transformation. This thesis group summarizes my results on the absolute pose estimation topic and two data fusion applications.

   (a) I experimentally tested the performance of the absolute pose estimation algorithm of omnidirectional cameras introduced in [Tamas, Frohlich, Kato, 2014] on synthetic data. For a common registration framework for central cameras I

implemented the proposed spherical surface integral calculation that reformulates [Tamas, Frohlich, Kato, 2014] to work with triangles of a mesh representation, and I deducted an efficient 2D geometric moments calculation scheme for the surface integrals of perspective cameras presented in [77]. I proposed an initialization step of the rotation and translation parameters for both spherical and perspective cameras, that works automatically using the projection of the corresponding 2D-3D regions. Through quantitative evaluation of the method, I proved its performance, I compared it to previous point-wise spherical integral approximation approach [Tamas, Frohlich, Kato, 2014] on large scale synthetic data, while also comparing the spherical and classical models applied for the perspective camera. I also demonstrated the performance and usability of the method on multiple real data test cases with different cameras and 3D sensors.

(b) For the first visual data fusion application for cultural heritage objects, I adapted our region-based registration method [Tamas, Frohlich, Kato, 2014] extending it to non-planar, smooth surfaces. As part of the workflow, I proposed an ICP refinement step based on intensity data edges, and a simple solution for the multi-camera fusion problem based on the cameras' orientation. I experimentally proved that despite the change to non-planar surfaces, the robustness of the method remains the same, while also conducting real tests on collected data of cultural heritage objects. The second application focuses on the selection of views from large number of cameras. I implemented a more complex camera selection algorithm, to fully benefit from the different focal length, resolution and position of cameras, based on multiple criteria, like visibility, sharpness, viewing angle and resolution. Visualizing the fusion results required a solution for the correct texture mapping between the 3D model and hundreds of texture image files, thus I proposed a technical solution that can easily use the original images as textures, without the need to create specially baked texture files. I validated the proposed pipeline on the acquired 2D-3D large scale dataset of two Reformed churches.

II. **Planar Homography Estimation and 3D Reconstruction**

The 2D registration framework of [78] can also be extended for estimating planar homographies between spherical cameras. Practically the homographies would act in this case between the spherical projections in the two cameras, representing the image of the same planar region. In general, relative pose parameters, as well as the normal and distance of the inducing plane can be factorized from such a planar homography, but due to the inherent parametrization of a planar homography, direct approaches for solving the problem are also possible, avoiding the factorization step completely. This thesis group summarizes my results on the planar homography estimation and 3D reconstruction topics.

(a) I experimentally validated the proposed region-based homography estimation method for omnidirectional cameras using two of the most commonly used models. Following [110] I deducted the decomposition of relative pose parameters from homographies assuming a weak Manhattan world constraint, then proved its comparable performance to the standard factorization method of [136] on synthetic data. If relative pose is available, one can also calculate

	I		II	
	a	b	a	b
[Tamas, Frohlich, Kato, 2014]	•			
[Frohlich, Tamas, Kato, 2019]	•			
[Frohlich *et al.*, 2016]		•		
[Frohlich *et al.*, 2018]		•		
[Frohlich, Tamas, Kato, 2016]			•	
[Molnár *et al.*, 2014]			•	
[Frohlich, Kato, 2018]				•

Table A.1. The connection between the thesis points and publications.

the parameters of the inducing planar patch from the homography. I validated the proposed differential geometric approach for the computation of the normal vector, using the homographies estimated by our method [Frohlich, Tamas, Kato, 2016]. Through comparative evaluation on synthetic data, I proved, that the proposed method outperforms the classical method of [15], and it is robust against noise in the rotation and translation parameters.

(b) Taking a different approach on the homography estimation problem with perspective cameras, a standard parametrization of the homography was applied through the relative pose and plane parameters. Each camera pair and each available region pair defines a new homography, thus I deducted the homography equations in a multi-camera multi-region setup through the common pose and plane parameters, and validated the algorithm both in a minimal case setup, and various configurations of cameras and regions. For the multi-camera setup I built a bundle adjustment to simultaneously estimate all the unknown parameters of the system. I experimentally proved the method's performance on synthetic and on real data with precise Lidar pointcloud and marker based measurements as reference, and also on the KITTI benchmark dataset where it proved State-of-the-Art performance in comparison to the point-based multi-view reconstruction method of [144, 145].

Appendix B

Summary in Hungarian

A *számítógépes látás* az a tudományterület, melynek célja a digitális képek elemzése által, különböző számítási eszközöket felhasználva, magasabb-rendű információkhoz jutni. A terület egyik alapvető feladata egy kamera világhoz képesti pozíciójának és orientációjának a meghatározása, vagyis egy referencia koordináta rendszerben kifejezett abszolút pose becslése. Ha rendelkezésünkre áll legalább két kamera-kép és azok helyzete egy közös koordináta rendszerben, lehetőségünk van direkt módon a képekről hiányzó mélységi információ rekonstruálására, hasonló elv alapján, mint ahogy az emberi látás érzékeli a mélységet. A pose becslés elengedhetetlen lépése bármely számítógépes látás algoritmusnak, míg a 3D rekonstrukció is gyakran használt lépés valós alkalmazásokban. Mivel a kutatásom célja az volt, hogy újszerű régió-alapú megoldásokat javasoljak az egyes alapvető problémákra, síkhomográfiák használata által egy újfajta megközelítésre is lehetőségem nyílt. A dolgozatban összefoglaltam a pose becsléssel és 3D rekonstrukcióval kapcsolatos kutatási eredményeim.

B.1. Az eredmények tézisszerű összefoglalása

A dolgozat eredményeit két fő téziscsoportban foglaltam össze, ahol az elsőben abszolút pose becslésével és vizuális adatok fúziójával foglalkozom, míg a másodikban kamerák közötti síkhomográfia becsléssel és 3D síkrekonstrukcióval. A téziscsoportok és az elfogadott publikációim közötti kapcsolatot a B.1 táblázatban prezentálom.

I. **Abszolút pose becslés és adatfúzió**

A [78] által bemutatott 2D regisztrációs módszer által inspirálva, [77] egy újféle régió alapú abszolút pose becslő megoldást javasolt, amely perspektív kamerák egy 3D tér-adathoz képesti helyzetét képes meghatározni egy általános 2D-3D regisztrációs megoldással, mindenféle kalibrációs minta vagy explicit pontmegfeleltetések használata nélkül. Ez az alap ötlet kibővíthető egy általános abszolút pose becslési keretrendszerré centrális szférikus kamerák számára, amely különböző, vizuális adatfúziós feladatokra alkalmazható. Az alap ötlet egy nem-lineáris egyenletrendszer konstruálása, melynek a megoldása direkt módon adja meg a keresett transzformáció paramétereit. Ez a tézis csoport az abszolút pose becslési és adat-fúziós témákban elért eredményeimet foglalja össze.

(a) Kísérleti úton kimutattam az omnidirekcionális kamerák számára [Tamas, Frohlich, Kato, 2014] által bevezetett régió alapú abszolút pose becslő algoritmus

teljesítményét. A centrális kamerák számára bemutatott általános regisztrációs keretrendszerhez validáltam a gömbfelszíni háromszöghálón dolgozó integrál számolót. A perspektív kamerák felszíni integráljainak számolására egy hatékony, 2D geometriai momentumok rekurzív felírásán alapuló számolási módszert vezettem be. A forgatási és eltolási paraméterek inicializálására egy automatikus megoldást javasoltam perspektív és omni kamerák számára. A módszert kvantitatív kiértékeltem szintetikus adathalmazokon, összehasonlítva a korábbi pont-alapú integrál közelítéses megoldással, és vizsgálva a perspektív kamerák esetében a szférikus és klasszikus modell használatát. A módszer használhatóságát különféle kamerákkal és 3D szenzorokkal rögzített valós adatokon is igazoltam.

(b) A kulturális örökségvédelmi objektumok vizuális-adat fúziójához adaptáltam a régió-alapú regisztrációs módszerünket [Tamas, Frohlich, Kato, 2014], kiterjesztve azt nem-sík, de sima régiókra. A javasolt munkafolyamat részeként egy intenzitás információból kinyerhető élekre támaszkodó ICP alapú finomítási lépést javasoltam, míg a több kamerából történő fúzióra egy egyszerű megoldást a kamerák orientációja alapján. Kvantitatív kiértékelés alapján bizonyítottam, hogy a módszer nem-sík felületekre kiterjesztve is robusztus marad, míg valós, kulturális örökségvédelmi szempontból érdekes tárgyakon is helyes eredményeket kaptam. A második alkalmazás egy nagyméretű kulturális örökségvédelmi objektumok (például templomok) dokumentálására szolgáló fúziós munkafolyamat, melyhez megoldást javasoltam a nagy mennyiségű nézőpont esetében felmerülő kamera szelekciós kérdésre, ami figyelembe veszi az egyes kamerák rálátását, élességét, betekintési szögét és felbontását. Az eredmények vizualizációjára javasoltam egy olyan technikai megoldást, amely képes nagy mennyiségű, különálló textúra-kép kezelésére. A munkafolyamatot két református templomról rögzített 2D-3D nagyméretű adathalmazon validáltam.

II. **Síkhomográfia becslés és 3D rekonstrukció**

A [78] 2D regisztrációs megoldás kiterjeszthető szférikus kamerák között ható síkhomográfiák becslésére is. Lényegében ez esetben a homográfiák az azonos sík régiónak megfelelő gömbfelszíni vetületek között értelmezhetőek. Általánosságban elmondható, hogy a relatív pose paraméterek és az indukáló sík paraméterei különböző módszerekkel faktorizálhatóak ki az így meghatározott homográfiából, de a síkhomográfia eredendő paraméterezésének köszönhetően direkt megoldásokra is lehetőség nyílik, ezáltal kikerülhető a faktorizálás, annak minden velejáró bizonytalanságával. Ez a tézis csoport a síkhomográfia becslés és 3D síkrekonstrukció témákban elért eredményeimet foglalja össze.

(a) Kísérleti úton igazoltam az omnidirekcionális kamerák között ható síkhomográfiák becslésére javasolt módszert, többféle szférikus modellt felhasználva. [110] által inspirálva bemutattam egy megoldást a relatív pose faktorizálására síkhomográfiából, amely vertikális irány ismeretében és egy *Manhattan világ* feltételezés mellett a [136] standard módszerhez mérhető pontosságot produkált szintetikus adatokon. Ha már a relatív pose paraméterei rendelkezésre állnak, az indukáló sík paramétereit is meghatározhatjuk a homográfiából. A bemutatott módszerünk [Frohlich, Tamas, Kato, 2016] által becsült homográfiákkal

	I		II	
	a	b	a	b
[Tamas, Frohlich, Kato, 2014]	•			
[Frohlich, Tamas, Kato, 2019]	•			
[Frohlich *et al.*, 2016]		•		
[Frohlich *et al.*, 2018]		•		
[Frohlich, Tamas, Kato, 2016]			•	
[Molnár *et al.*, 2014]			•	
[Frohlich, Kato, 2018]				•

B.1. táblázat. A tézispontokhoz kapcsolódó publikációk.

igazoltam a sík normálvektorának a kiszámolására javasolt differenciál geometriai megoldást. Szintetikus adathalmazon végeztem összehasonlító kiértékelést, melyben a klasszikus [15] módszernél jobb teljesítményt értünk el, és a forgatási és eltolási paraméterekben levő hibákra is kellően robusztusnak bizonyult a módszerünk.

(b) A homográfia becslés az egyenletek megfelelő átparaméterezésével akár direkt módon is megadhatja a kamerák relatív pose-át és a sík paramétereit, ezzel kiküszöbölhető a homográfia faktorizálás és az azzal járó bizonytalanságok. Mivel minden kamera pár és minden régió egy újabb homográfiát határoz meg, ezek egyenleteit felírtam a közös paraméterek függvényében egy több kamerás több régiós rendszerben, és validáltam az algoritmust úgy a minimális megoldási esetben, mint több különböző konfigurációban is. A több kamerás esetre egy kötegelt behangolási megoldást is javasoltam, mely szimultán módon egyszerre finomítja az összes keresett paramétert. Az algoritmus teljesítményét kiértékeltem szintetikus és többféle valós adaton is: pontos Lidar pontfelhővel és marker alapú referencia mérésekkel rendelkező saját adathalmazon, továbbá a KITTI publikus adathalmazon is, ahol a módszerünk a legjobb pont-alapú általános rekonstrukciós módszernél [144] jobban teljesített.

Publications

Articles

[Frohlich, Tamas, Kato, 2019] R. Frohlich, L. Tamas, and Z. Kato. “Absolute Pose Estimation of Central Cameras Using Planar Regions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). accepted subject to minor revision, under review, pp. 1–16.

Book Chapters

[Frohlich *et al.*, 2018] R. Frohlich, S. Gubo, A. Lévai, and Z. Kato. “3D-2D Data Fusion in Cultural Heritage Applications”. In: *Heritage Preservation: A Computational Approach*. Ed. by B. Chanda, S. Chaudhuri, and S. Chaudhury. Springer Singapore, 2018, pp. 111–130. ISBN: 978-981-10-7221-5.

[Frohlich, Tamas, Kato, 2016] R. Frohlich, L. Tamas, and Z. Kato. “Handling Uncertainty and Networked Structure in Robot Control”. In: vol. 42. Studies in Systems, Decision and Control. Chapter 6. Springer, Feb. 2016. Chap. Homography Estimation Between Omnidirectional Cameras Without Point Correspondences, pp. 129–151.

Conference Papers

[Frohlich, Kato, 2018] R. Frohlich and Z. Kato. “Simultaneous Multi-View Relative Pose Estimation and 3D Reconstruction from Planar Regions”. In: *Proceedings of ACCV Workshop on Advanced Machine Vision for Real-life and Industrially Relevant Applications*. Ed. by G. Carneiro. Vol. 11367. Lecture Notes in Computer Science. Springer, Dec. 2018. ISBN: ISBN 978-3-030-21074-8.

[Frohlich *et al.*, 2016] R. Frohlich, Z. Kato, A. Tremeau, L. Tamas, S. Shabo, and Y. Waksman. “Region Based Fusion of 3D and 2D Visual Data for Cultural Heritage Objects”. In: *Proceedings of International Conference on Pattern Recognition*. IEEE. Cancun, Mexico: IEEE, Dec. 2016, pp. 2404–2409.

[Molnár *et al.*, 2014] J. Molnár, R. Frohlich, C. Dmitry, and Z. Kato. “3D Reconstruction of Planar Patches Seen by Omnidirectional Cameras”. In: *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*. Wollongong, Australia: IEEE, Nov. 2014, pp. 1–8. ISBN: ISBN 978-1-4799-5409-4.

[Tamas, Frohlich, Kato, 2014] L. Tamas, R. Frohlich, and Z. Kato. “Relative Pose Estimation and Fusion of Omnidirectional and Lidar Cameras”. In: *Proceedings of the ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*. Ed. by L. de Agapito, M. M. Bronstein, and C. Rother. Vol. 8926. Lecture Notes in Computer Science. Zurich, Switzerland: Springer, Sept. 2014, pp. 640–651. ISBN: ISBN 978-3-319-16180-8.

Bibliography

[1] M. David. *Vision*. W. H. Freeman & Company, 1982. ISBN: 0-7167-1284-9.

[2] F. Olivier. *Three-Dimensional Computer Vision, A Geometric Viewpoint*. MIT Press, 1993. ISBN: 978-0-262-06158-2.

[3] L. Kneip, D. Scaramuzza, and R. Siegwart. "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation". In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. IEEE, June 2011.

[4] Y. Yagi and S. Kawato. "Panorama scene analysis with conic projection". In: *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. July 1990, 181–187 vol.1.

[5] S. Baker and S. K. Nayar. "A Theory of Single-Viewpoint Catadioptric Image Formation". In: *International Journal of Computer Vision* 35.2 (1999), pp. 175–196.

[6] C. Geyer and K. Daniilidis. "A unifying theory for central panoramic systems". In: *European Conference on Computer Vision (ECCV)*. 2000, pp. 445–462.

[7] B. Mičušık and T. Pajdla. "Para-catadioptric Camera Auto-calibration from Epipolar Geometry". In: *Proc. of the Asian Conference on Computer Vision (ACCV)* (Jeju Island, Korea South). Ed. by K.-S. Hong and Z. Zhang. Vol. 2. Seoul, Korea South: Asian Federation of Computer Vision Societies, Jan. 2004, pp. 748–753. ISBN: 89-954842-0-9.

[8] L. Puig, Y. Bastanlar, et al. "Calibration of Central Catadioptric Cameras Using a DLT-Like Approach". In: *International Journal of Computer Vision* 93.1 (May 2011), pp. 101–114.

[9] D. Scaramuzza, A. Martinelli, and R. Siegwart. "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion". In: *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*. ICVS-06. Washington, USA: IEEE Computer Society, 2006, pp. 45–51.

[10] P. Sturm, S. Ramalingam, et al. "Camera Models and Fundamental Concepts Used in Geometric Computer Vision". In: *Foundations and Trends in Computer Graphics and Vision* 6.1-2 (Jan. 2011), pp. 1–183.

[11] X. Ying and Z. Hu. "Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model". In: *Proceedings of European Conference on Computer Vision*. Springer Berlin Heidelberg, 2004, pp. 442–455.

[12] J. Barreto and H. Araujo. "Issues on the geometry of central catadioptric image formation". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, 2001.

[13] B. Micusik and T. Pajdla. “Structure from motion with wide circular field of view cameras”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.7 (July 2006), pp. 1135–1149.

[14] D. Scaramuzza, A. Martinelli, and R. Siegwart. “A Toolbox for Easily Calibrating Omnidirectional Cameras.” In: *IEEE/RSJ International Conference on Intelligent Robots*. Bejing: IEEE, Oct. 2006, pp. 5695–5701.

[15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, UK: Cambridge University Press, 2004.

[16] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (1981), pp. 381–395.

[17] S. A. K. Tareen and Z. Saleem. “A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK”. In: *Proceedings of International Conference on Computing, Mathematics and Engineering Technologies*. IEEE, Mar. 2018.

[18] S. Krüger and A. Calway. “Image Registration using Multiresolution Frequency Domain Correlation”. In: *Procedings of British Machine Vision Conference*. British Machine Vision Association, 1998.

[19] P. Sturm and W. Triggs. “A Factorization Based Algorithm for multi-Image Projective Structure and Motion”. In: *Proceedings of European Conference on Computer Vision*. Ed. by B. Buxton and R. Cipolla. Springer, 1996.

[20] D. Nistér, O. Naroditsky, and J. Bergen. “Visual odometry”. In: *Computer Vision and Pattern Recognition*. Vol. 1. IEEE. Washington, DC, USA, June 2004, pp. 1–8.

[21] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. “Camera-IMU-based localization: Observability analysis and consistency improvement”. In: *The International Journal of Robotics Research* 33.1 (2014), pp. 182–201.

[22] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. “Real-time Self-localization from Panoramic Images on Mobile Devices”. In: *Proceedings of International Symposium on Mixed and Augmented Reality*. Basel, Switzerland: IEEE Computer Society, Oct. 2011, pp. 37–46.

[23] A. Geiger, M. Lauer, et al. “3D Traffic Scene Understanding From Movable Platforms”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.5 (2014), pp. 1012–1025.

[24] D. Lin, S. Fidler, and R. Urtasun. “Holistic Scene Understanding for 3D Object Detection with RGBD Cameras”. In: *International Conference on Computer Vision, Sydney, Australia*. IEEE Computer Society, Dec. 2013, pp. 1417–1424.

[25] P. T. Furgale, U. Schwesinger, et al. “Toward automated driving in cities using close-to-market sensors: An overview of the V-Charge Project”. In: *Intelligent Vehicles Symposium*. Gold Coast City, Australia, June 2013, pp. 809–816.

[26] D. F. Dementhon and L. S. Davis. “Model-based object pose in 25 lines of code”. In: *International Journal of Computer Vision* 15.1-2 (June 1995), pp. 123–141.

[27] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. “Object Pose: The Link between Weak Perspective, Paraperspective, and Full Perspective”. In: *International Journal of Computer Vision* 22.2 (1997), pp. 173–189.

[28] V. Lepetit, F. Moreno-Noguer, and P. Fua. “EPnP: An Accurate O(n) Solution to the PnP Problem”. In: *Int. J. Comput. Vision* 81.2 (Feb. 2009), pp. 155–166. ISSN: 0920-5691.

[29] S. Li, C. Xu, and M. Xie. “A Robust O(n) Solution to the Perspective-n-Point Problem”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7 (July 2012), pp. 1444–1450.

[30] L. Kneip, H. Li, and Y. Seo. “UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability”. In: *Proceedings of European Conference on Computer Vision*. Vol. 8689. Lecture Notes in Computer Science. Zurich, Switzerland: Springer, Sept. 2014, pp. 127–142.

[31] C. Xu, L. Zhang, L. Cheng, and R. Koch. “Pose Estimation from Line Correspondences: A Complete Analysis and A Series of Solutions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2016), pp. 1209–1222. ISSN: 0162-8828.

[32] N. Horanyi and Z. Kato. “Generalized Pose Estimation from Line Correspondences with Known Vertical Direction”. In: *International Conference on 3D Vision*. Qingdao, China: IEEE, Oct. 2017, pp. 244–253.

[33] N. Snavely, S. M. Seitz, and R. Szeliski. “Photo Tourism: Exploring Photo Collections in 3D”. In: *ACM SIGGRAPH*. Boston, Massachusetts: ACM, 2006, pp. 835–846. ISBN: 1-59593-364-6.

[34] F. Camposeco, T. Sattler, and M. Pollefeys. “Minimal Solvers for Generalized Pose and Scale Estimation from Two Rays and One Point”. In: *Proceedings of European Conference Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9909. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 202–218.

[35] G. H. Lee. “A Minimal Solution for Non-perspective Pose Estimation from Line Correspondences”. In: *Proceedings of European Conference on Computer Vision*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 170–185.

[36] J. Kannala and S. S. Brandt. “A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.8 (2006), pp. 1335–1340.

[37] S. K. Nayar. “Catadioptric Omnidirectional Camera”. In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR ’97)*. CVPR ’97. Washington, USA: IEEE Computer Society, 1997, pp. 482–. ISBN: 0-8186-7822-4.

[38] M. Schoenbein, T. Strauss, and A. Geiger. “Calibrating and Centering Quasi-Central Catadioptric Cameras”. In: *International Conference on Robotics and Automation*. Hong-Kong, June 2014, pp. 1253–1256.

[39] C. Mei and P. Rives. “Single View Point Omnidirectional Camera Calibration from Planar Grids”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Roma, Italy, Apr. 2007.

[40] D. Scaramuzza, A. Harati, and R. Siegwart. “Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes”. In: *IEEE International Conference on Intelligent Robots and Systems*. IEEE/RSJ. San Diego, USA: IEEE, Oct. 2007, pp. 4164–4169.

[41] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis. “3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization”. In: *The International Journal of Robotics Research* 31.4 (2012), pp. 452–467.

[42] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice. “Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information”. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*. Toronto, Canada, July 2012, pp. 2053–2059.

[43] N. Schneider, F. Piewak, C. Stiller, and U. Franke. “RegNet: Multimodal sensor registration using deep neural networks”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 1803–1810.

[44] G. Iyer, K. Ram R., J. Krishna Murthy, and K. Madhava Krishna. “CalibNet: Self-Supervised Extrinsic Calibration using 3D Spatial Transformer Networks”. In: *ArXiv e-prints, 1803.08181* (Mar. 2018). arXiv: 1803.08181 [cs.RO].

[45] Z. Taylor and J. Nieto. “A Mutual Information Approach to Automatic Calibration of Camera and Lidar in Natural Environments”. In: *Australian Conference on Robotics and Automation*. Wellington, Australia, Dec. 2012, pp. 3–5.

[46] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. “Mutual-information-based registration of medical images: a survey”. In: *IEEE Transcations on Medical Imaging* (2003), pp. 986–1004.

[47] D. Paudel, C. Demonceaux, A. Habed, and P. Vasseur. “Localization of 2D Cameras in a Known Environment Using Direct 2D-3D Registration”. In: *Pattern Recognition (ICPR), 2014 22nd International Conference on*. Aug. 2014, pp. 196–201.

[48] A. Banno and K. Ikeuchi. “Omnidirectional Texturing Based on Robust 3D Registration Through Euclidean Reconstruction from Two Spherical Images”. In: *Computer Vision Image Understanding* 114.4 (2010), pp. 491–499.

[49] A. Mastin, J. Kepner, and J. W. F. III. “Automatic Registration of LIDAR and Optical Images of Urban Scenes”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Miami, Florida, USA: IEEE, June 2009, pp. 2639–2646.

[50] S. Bileschi. “Fully automatic calibration of lidar and video streams from a vehicle”. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 1457–1464.

[51] M. Corsini, M. Dellepiane, et al. “Fully Automatic Registration of Image Sets on Approximate Geometry”. In: *International Journal of Computer Vision* 102.1-3 (2013), pp. 91–111. ISSN: 0920-5691.

[52] T. Franken, M. Dellepiane, et al. “Minimizing user intervention in registering 2D images to 3D models.” In: *The Visual Computer* 21.8-10 (2005), pp. 619–628.

[53] H. S. Alismail, L. D. Baker, and B. Browning. “Automatic Calibration of a Range Sensor and Camera System”. In: *Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission*. Zurich, Switzerland: IEEE, Oct. 2012, pp. 286–292.

[54] O. Naroditsky, E. P. Iv, and K. Daniilidis. “Automatic Alignment of a Camera with a Line Scan Lidar System”. In: *International Conference on Robotics and Automation*. Shanghai, China: IEEE, May 2011, pp. 3429–3434.

[55] Y. Liu, T. Huang, and O. Faugeras. “Determination of camera location from 2D to 3D line and point correspondences”. In: *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR ’88., Computer Society Conference on*. June 1988, pp. 82–88.

[56] L. Liu and I. Stamos. “Automatic 3D to 2D Registration for the Photorealistic Rendering of Urban Scenes.” In: *Conference on Computer Vision and Pattern Recognition*. Vol. 2. IEEE Computer Society, June 2005, pp. 137–143.

[57] R. Unnikrishnan and M. Hebert. *Fast Extrinsic Calibration of a Laser Rangefinder to a Camera*. Tech. rep. Carnegie Mellon University, 2005.

[58] Q. Zhang. “Extrinsic Calibration of a Camera and Laser Range Finder”. In: *International Conference on Intelligent Robots and Systems*. Sendai, Japan: IEEE, Sept. 2004, pp. 2301–2306.

[59] D. Herrera C, J. Kannala, and J. Heikkila. “Joint Depth and Color Camera Calibration with Distortion Correction.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.10 (2012), pp. 1–8.

[60] J. Levinson and S. Thrun. “Automatic Online Calibration of Cameras and Lasers”. In: *Proceedings of Robotics: Science and Systems*. Berlin, Germany, June 2013.

[61] H. P. A. Lensch and W. Heidrich. “A Silhouette-Based Algorithm for Texture Registration and Stitching”. In: *Graphical Models* 63 (2001), pp. 245–262.

[62] P. Nunez, P. Drews, R. Rocha, and J. Dias. “Data Fusion Calibration for a 3D Laser Range Finder and a Camera using Inertial Data”. In: *European Conference on Mobile Robots*. Dubrovnik, Croatia, Sept. 2009, pp. 31–36.

[63] Z. Taylor, J. Nieto, and D. Johnson. “Multi-Modal Sensor Calibration Using a Gradient Orientation Measure”. In: *Journal of Field Robotics* 32.5 (2015), pp. 675–695.

[64] D. G. Lowe. “Fitting Parameterized Three-Dimensional Models to Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991), pp. 441–450.

[65] A. Taneja, L. Ballan, and M. Pollefeys. “Registration of Spherical Panoramic Images with Cadastral 3D Models”. In: *Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. 3DIMPVT. Washington, DC, USA: IEEE Computer Society, 2012, pp. 479–486.

[66] M. Zhang, Y. Chen, and G. Wang. “Hybrid Silhouette and Key-Point Driven 3D-2D Registration”. In: *Image and Graphics (ICIG), 2013 Seventh International Conference on*. July 2013, pp. 585–590.

[67] H. Rüther, M. Chazan, et al. “Laser scanning for conservation and research of African cultural heritage sites: the case study of Wonderwerk Cave, South Africa”. In: *Journal of Archaeological Science* 36.9 (2009), pp. 1847–1856. ISSN: 0305-4403.

[68] C. Santagati, L. Inzerillo, and F. D. Paola. “Image-Based Modeling Techniques for Architectural Heritage 3D Digitalization: Limits and Potentialities”. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-5.w2 (Sept. 2013), pp. 555–560.

[69] R. Kadobayashi, N. Kochi, H. Otani, and R. Furukawa. “Comparison and evaluation of laser scanning and photogrammetry and their combined use for digital recording of cultural heritage”. In: *Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 35.5 (2004), pp. 401–406.

[70] L. Grosman, O. Smikt, and U. Smilansky. “On the application of 3-D scanning technology for the documentation and typology of lithic artifacts”. In: *Journal of Archaeological Science* 35.12 (2008), pp. 3101–3110. ISSN: 0305-4403.

[71] A. Koutsoudis, B. Vidmar, and F. Arnaoutoglou. “Performance evaluation of a multi-image 3D reconstruction software on a low-feature artefact”. In: *Journal of Archaeological Science* 40.12 (2013), pp. 4450–4456. ISSN: 0305-4403.

[72] P. Grussenmeyer, T. Landes, T. Voegtle, and K. Ringle. “Comparison methods of terrestrial laser scanning, photogrammetry and tacheometry data for recording of cultural heritage buildings”. In: *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 37(B5) (2008), pp. 213–218.

[73] S. Al-kheder, Y. Al-shawabkeh, and N. Haala. “Developing a documentation system for desert palaces in Jordan using 3D laser scanning and digital photogrammetry”. In: *Journal of Archaeological Science* 36.2 (2009), pp. 537–546. ISSN: 0305-4403.

[74] F. Agnello and M. L. Brutto. “Integrated surveying techniques in cultural heritage documentation”. In: *ISPRS Archives* 36 (2007), p. 5.

[75] S. El-Hakim, L. Gonzo, et al. “Detailed 3D Modelling of Castles:” in: *International journal of architectural computing : IJAC*. International journal of architectural computing : IJAC 5.2 (2007), pp. 200–220.

[76] P. Viola and W. M. Wells III. “Alignment by Maximization of Mutual Information”. In: *International Journal of Computer Vision* 24.2 (Sept. 1997), pp. 137–154. ISSN: 0920-5691.

[77] L. Tamas and Z. Kato. “Targetless Calibration of a Lidar - Perspective Camera Pair”. In: *Proceedings of ICCV Workshop on Big Data in 3D Computer Vision*. IEEE. Sydney, Australia: IEEE, Dec. 2013, pp. 668–675.

[78] C. Domokos, J. Nemeth, and Z. Kato. “Nonlinear Shape Registration without Correspondences”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (May 2012), pp. 943–958. ISSN: 0162-8828.

[79] J. E. G. Joseph O’Rourke, ed. *Handbook of Discrete and Computational Geometry, Second Edition (Discrete Mathematics and Its Applications)*. Chapman and Hall/CRC, 2004. ISBN: 9781584883012.

[80] J. M. Pozo, M. C. Villa-Uriol, and A. F. Frangi. “Efficient 3D Geometric and Zernike Moments Computation from Unstructured Surface Meshes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3 (Mar. 2011), pp. 471–484. ISSN: 0162-8828.

[81] P. Koehl. “Fast Recursive Computation of 3D Geometric Moments from Surface Meshes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (Nov. 2012), pp. 2158–2163. ISSN: 0162-8828.

[82] J.-G. Leu. “Computing a shape’s moments from its boundary”. In: *Pattern Recognition* 24.10 (1991), pp. 949–957. ISSN: 0031-3203.

[83] M. H. Singer. “A general approach to moment calculation for polygons and line segments”. In: *Pattern Recognition* 26.7 (1993), pp. 1019–1028. ISSN: 0031-3203.

[84] X. Jiang and H. Bunke. “Simple and fast computation of moments”. In: *Pattern Recognition* 24.8 (1991), pp. 801–806. ISSN: 0031-3203.

[85] G. C. Best. “Helpful Formulas for Integrating Polynomials in Three Dimensions (in Technical Notes and Short Papers)”. In: *International Journal of Mathematics and Computer Science* 18.86 (Apr. 1964), pp. 310–312.

[86] P.-O. Persson and G. Strang. “A simple mesh generator in MATLAB”. In: *SIAM review* 46.2 (2004), pp. 329–345.

[87] S. R. Vantaram and E. Saber. “Survey of contemporary trends in color image segmentation”. In: *Journal of Electronic Imaging* 21.4 (2012), pp. 1–28.

[88] M. Preetha, L. Suresh, and M. Bosco. “Image segmentation using seeded region growing”. In: *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*. 2012, pp. 576–583.

[89] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan. “Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds”. In: *ArXiv e-prints* (Sept. 2012).

[90] A. Nurunnabi, D. Belton, and G. West. “Robust Segmentation in Laser Scanning 3D Point Cloud Data”. In: *Proc. of Int. Conf. on Digital Image Computing: Techniques and Applications*. Dec. 2012, pp. 1–8.

[91] L. Tamas and L. C. Goron. “3D semantic interpretation for robot perception inside office environments”. In: *Eng. Appl. of AI* 32 (2014), pp. 76–87.

[92] N. Akkiraju and H. Edelsbrunner. “Triangulating the Surface of a Molecule”. In: *Discrete Applied Mathematicss* 71.1-3 (Dec. 1996), pp. 5–22.

[93] O. Józsa, A. Börcs, and C. Benedek. “Towards 4D Virtual City Reconstruction From Lidar Point Cloud Sequences”. In: *ISPRS Workshop on 3D Virtual City Modeling*. Vol. II-3/W1. ISPRS Annals of Photogrammetry, Remote Sensing and the Spatial Information Sciences. Regina, Canada, 2013, pp. 15–20.

[94] L. Tamas and A. Majdik. “Heterogeneous Feature Based Correspondence Estimation”. In: *Multisensor Fusion and Integration for Intelligent Systems*. IEEE. Hamburg, Germany, June 2012, pp. 89–94.

[95] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* (2013), pp. 1231–1237.

[96] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref. “Automatic image segmentation by integrating color-edge extraction and seeded region growing”. In: *IEEE Trans. on Image Processing* 10.10 (Oct. 2001), pp. 1454–1466. ISSN: 1057-7149.

[97] D. Sedlacek and J. Zara. “Graph Cut Based Point-Cloud Segmentation for Polygonal Reconstruction”. In: *Proc. of the 5th Int. Symposium on Advances in Visual Computing: Part II*. ISVC ’09. Las Vegas, Nevada: Springer-Verlag, 2009, pp. 218–227. ISBN: 978-3-642-10519-7.

[98] P. J. Besl and N. D. McKay. “Method for registration of 3-D shapes”. In: *Robotics-DL tentative*. International Society for Optics and Photonics. 1992, pp. 586–606.

[99] J.-Y. Bouguet. “Caltech calibration toolbox”. In: ().

[100] S. Katz, A. Tal, and R. Basri. “Direct Visibility of Point Sets”. In: *ACM SIGGRAPH 2007 Papers*. SIGGRAPH ’07. San Diego, California: ACM, 2007.

[101] G. Yang and B. J. Nelson. “Wavelet Based Autofocusing and Unsupervised Segmentation of Microscopic Images”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003, Las Vegas, Nevada, October*. Vol. 3. IEEE, 2003, 2143–2148 vol.3.

[102] D. OuYang and H.-Y. Feng. “On the Normal Vector Estimation for Point Cloud Data from Smooth Surfaces”. In: *Comput. Aided Des.* 37.10 (Sept. 2005), pp. 1071–1079. ISSN: 0010-4485.

[103] M. Garland and P. S. Heckbert. “Surface simplification using quadric error metrics”. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’97* (1997), pp. 209–216. ISSN: 00978930.

[104] K. McHenry and P. Bajcsy. *An overview of 3D data content, file formats and viewers*. Tech. rep. 2008, p. 21.

[105] Y.-C. Chang and J. F. Reid. “RGB calibration for color image analysis in machine vision”. In: *IEEE Transactions on Image Processing* 5.10 (Oct. 1996), pp. 1414–1422. ISSN: 1057-7149.

[106] P. Sturm. “Algorithms for plane-based pose estimation”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. Vol. 1. June 2000, pp. 706–711.

[107] C. Mei, S. Benhimane, E. Malis, and P. Rives. “Efficient Homography-Based Tracking and 3-D Reconstruction for Single-Viewpoint Sensors”. In: *Robotics, IEEE Transactions on* 24.6 (Dec. 2008), pp. 1352–1364. ISSN: 1552-3098.

[108] G. Caron, E. Marchand, and E. M. Mouaddib. “Tracking planes in omnidirectional stereovision.” In: *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 6306–6311.

[109] A. Makadia, C. Geyer, and K. Daniilidis. “Correspondence-free Structure from Motion”. English. In: *International Journal of Computer Vision* 75.3 (Dec. 2007), pp. 311–327. ISSN: 0920-5691.

[110] O. Saurer, F. Fraundorfer, and M. Pollefeys. “Homography based visual odometry with known vertical direction and weak Manhattan world assumption”. In: *IEEE/IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)*. 2012.

[111] J. Molnár, R. Huang, and Z. Kato. “3D Reconstruction of Planar Surface Patches: A Direct Solution”. In: *Proceed ings of ACCV Workshop on Big Data in 3D Computer Vision*. Ed. by C. V. Jawahar and S. Shan. Vol. 9008. Lecture Notes in Computer Science. Singapore: Springer, Nov. 2014, pp. 286–300.

[112] A. Tanács, A. Majdik, et al. “Collaborative Mobile 3D Reconstruction of Urban Scenes”. In: *Proceedings of ACCV Workshop on Intelligent Mobile and Egocentric Vision*. Ed. by C. V. Jawahar and S. Shan. Vol. 9010. Lecture Notes in Computer Science. Singapore: Springer, Nov. 2014, pp. 486–501.

[113] K. Mikolajczyk, T. Tuytelaars, et al. “A Comparison of Affine Region Detectors”. In: *Int. J. Comput. Vision* 65.1-2 (Nov. 2005), pp. 43–72.

[114] Y. Furukawa and J. Ponce. “Accurate, Dense, and Robust Multiview Stereopsis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010), pp. 1362–1376.

[115] A. Tanács, A. Majdik, et al. “Establishing Correspondences between Planar Image Patches”. In: *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*. Best Paper Award. Wollongong, Australia: IEEE, Nov. 2014, pp. 1–7.

[116] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[117] L. Puig and J. J. Guerrero. “Scale space for central catadioptric systems: Towards a generic camera feature extractor”. In: *Proceedings of International Conference on Computer Vision*. IEEE. 2011, pp. 1599–1606.

[118] D. Gutierrez, A. Rituerto, J. Montiel, and J. Guerrero. “Adapting a real-time monocular visual SLAM from conventional to omnidirectional cameras”. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. Nov. 2011, pp. 343–350.

[119] T. Svoboda and T. Pajdla. “Epipolar Geometry for Central Catadioptric Cameras”. In: *International Journal of Computer Vision* 49.1 (2002), pp. 23–37.

[120] R. Basri and D. W. Jacobs. “Recognition using region correspondences”. In: *International Journal of Computer Vision* 25 (1996), pp. 141–162.

[121] M. Habbecke and L. Kobbelt. “Iterative Multi-View Plane Fitting”. In: *In VMV’06*. 2006, pp. 73–80.

[122] M. Habbecke and L. Kobbelt. “A Surface-Growing Approach to Multi-View Stereo Reconstruction”. In: *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on* (2007), pp. 1–8.

[123] S. Sinha, D. Steedly, and R. Szeliski. “Piecewise planar stereo for image-based rendering”. In: *Computer Vision, 2009 IEEE 12th International Conference on* (2009), pp. 1881–1888.

[124] A. Kowdle, Y.-J. Chang, A. Gallagher, and T. Chen. “Active learning for piecewise planar 3D reconstruction”. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 929–936. ISBN: 978-1-4577-0394-2.

[125] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. “Towards high-resolution large-scale multi-view stereo”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), pp. 1430–1437.

[126] F. Fraundorfer, K. Schindler, and H. Bischof. “Piecewise planar scene reconstruction from sparse correspondences”. In: *Image Vision Comput.* 24.4 (Apr. 2006), pp. 395–406.

[127] Z. Zhou, H. Jin, and Y. Ma. “Robust plane-based structure from motion”. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1482–1489. ISBN: 978-1-4673-1226-4.

[128] P. Musialski, P. Wonka, et al. “A Survey of Urban Reconstruction”. In: *EUROGRAPHICS 2012 State of the Art Reports*. Eurographics Association, 2012, pp. 1–28.

[129] B. Micusik and J. Kosecka. “Piecewise planar city 3D modeling from street view panoramic sequences”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. IEEE, June 2009.

[130] J. Levinson, J. Askeland, et al. “Towards fully autonomous driving: Systems and algorithms”. In: *Proceedings of Intelligent Vehicles Symposium*. IEEE, June 2011.

[131] H. S. Lee and K. Kim. “Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.5 (May 2018), pp. 1652–1663.

[132] Á. Arcos-Garcıa, J. A. Álvarez-Garcıa, and L. M. Soria-Morillo. “Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods”. In: *Neural Networks* 99 (Mar. 2018), pp. 158–165.

[133] A. Martinović, M. Mathias, J. Weissenberg, and L. V. Gool. “A Three-Layered Approach to Facade Parsing”. In: *Proceedings of European Conference on Computer Vision*. Springer, 2012, pp. 416–429.

[134] O. Faugeras and F. Lustman. *Motion and structure from motion in a piecewise planar environment*. Tech. rep. RR-0856. Sophia Antipolis, France: INRIA, June 1988.

[135] M. Recky and F. Leberl. “Window detection in complex facades”. In: *Proceedings of European Workshop on Visual Information Processing*. 2010, pp. 220–225.

[136] O. Faugeras and F. Lustman. *Motion and structure from motion in a piecewise planar environment*. Tech. rep. RR-0856. June 1988.

[137] J. Coughlan and A. L. Yuille. “Manhattan World: compass direction from a single image by Bayesian inference”. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. 1999, pp. 941–947.

[138] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. “Manhattan-world stereo”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 1422–1429. ISBN: 978-1-4244-3992-8.

[139] F. Devernay and O. Faugeras. “Computing differential properties of 3-D shapes from stereoscopic images without 3-D models”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. June 1994, pp. 208–213.

[140] D. G. Jones and J. Malik. “Determining three-dimensional shape from orientation and spatial frequency disparities”. In: *Proceedings of European Conference on Computer Vision*. Ed. by G. Sandini. Vol. 588. Lecture Notes in Computer Science. Springer, 1992, pp. 661–669.

[141] J. Molnár and D. Chetverikov. “Quadratic Transformation for Planar Mapping of Implicit Surfaces”. In: *Journal of Mathematical Imaging and Vision* 48.1 (2014), pp. 176–184.

[142] A. Geiger, P. Lenz, and R. Urtasun. “Are we ready for autonomous driving? The KITTI vision benchmark suite”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. IEEE, June 2012.

[143] A. Geiger, J. Ziegler, and C. Stiller. “StereoScan: Dense 3d reconstruction in real-time”. In: *Proceedings of Intelligent Vehicles Symposium*. IEEE, June 2011.

[144] J. L. Schonberger and J.-M. Frahm. “Structure-from-Motion Revisited”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. IEEE, June 2016.

[145] J. L. Schonberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. “Pixelwise View Selection for Unstructured Multi-View Stereo”. In: *Proceedings of European Conference on Computer Vision*. Springer, 2016, pp. 501–518.

[146] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. “Tanks and temples”. In: *ACM Transactions on Graphics* 36.4 (July 2017), pp. 1–13.