TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DISSERTATION

Motion Estimation from Integration of Range and Intensity Data

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

Univ.Prof. Dipl.Ing. Dr.techn. Norbert Pfeifer

E120.7

Forschungsgruppe Photogrammetrie

Department für Geodäsie und Geoinformation

eingereicht an der Technischen Universität Wien

Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

von

Sajid Ghuffar, MSc.

Matrik. Nr. 0929210

Peter Jordan Strasse 1/W012,1190 Wien

Wien, 27.05.2014

(Sajid Ghuffar, MSc.)

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Motion Estimation from Integration of Range and Intensity Data

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor/in der technischen Wissenschaften

by

Sajid Ghuffar, MSc.
Registration Number 0929210

to the Faculty of Mathematics and Geoinformation
at the Vienna University of Technology

Advisor: Univ.Prof. Dipl.Ing. Dr.techn. Norbert Pfeifer

The dissertation has been reviewed by:

(Prof. Norbert Pfeifer) (Prof. Konrad Schindler)

Vienna, 27.05.2014

(Sajid Ghuffar, MSc.)

Erklärung zur Verfassung der Arbeit

Sajid Ghuffar, MSc.
Peter Jordan Strasse 1/W012, 1190 Vienna

I hereby declare, that I independently drafted this manuscript, that all sources and references used are correctly cited and that the respective parts of this manuscript - including tables, maps and figures - which were included from other manuscripts or the internet, either semantically or syntactically, are made clearly evident in the text and all respective sources are correctly cited.

_____ _____

(Ort, Datum) (Unterschrift Verfasser)

Erklärung zur Verfassung der Arbeit

Sajid Ghuffar, MSc.
Peter Jordan Strasse 1/W012, 1190 Wien

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

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgements

First of all, I would like to thank my supervisor Norbert Pfeifer for his support, advices and kindness throughout this research work and being very encouraging during all this time. Then, I want to thank Konrad Schindler for examining my thesis and providing a thorough review in a short amount of time. I also thank Margrit Gelautz, as being the head of the *Doctoral College on Computational Perception* for supporting this research work.

I thank all my colleagues at the research group Photogrammetry, Department of Geodesy and Geoinformation for their kind support and cooperation during the course of this research work. I really appreciate the efforts of Camillo Ressl, Helmut Kager, Wilfried Karel, Markus Hollaus, Johannes Otepka and Josef Jansa for reviewing my work and teaching me various topics. I thank Ewelina Rupnik, Milutin Milenkovic and Ana Djuricic for their help and support throughout this PhD and sharing so many memorable moments of these years. I thank my office mates Andreas Roncat and Balazs Szekely for providing a very friendly and cooperative office environment. I also thank Nicole Brosch for her cooperation in proposing and writing the joint papers. Furthermore, I thank Sabine Horvath, Sascha Rasztovits, Philipp Glira, Michael Hornacek, Zeeshan Zia and Rameez Naqvi for their suggestions. I am also thankful to Hans Thüminger for being so helpful in all IT related matters.

Finally, I thank my parents and my family members for their support and patience during my PhD studies.

Major part of this research work is supported by the *Doctoral College on Computational Perception* at Vienna University of Technology. Further support came from the following projects: *High Performance Computational Intelligence Methoden zur Auswertung von Airborne Laser Scanning Daten* funded by the Austrian Research Promotion Agency (FFG) and the Alpine Space project *NEWFOR*

Abstract

In recent years, low cost, high frame rate 3D or range cameras, which simultaneously provide distance and intensity information have become commercially available. These cameras have gained a lot of interest in numerous applications. Indoor mapping and autonomous mobile navigation are two example applications, which have a lot of potential for such cameras. Motion estimation is an integral part of these applications. Therefore, it is vital to investigate methods and techniques for motion estimation, which can exploit the simultaneous availability of range and intensity information provided by these 3D cameras.

This thesis investigates the integration of range and intensity data for the task of motion estimation. The motion estimation of a moving camera and motion estimation of independently moving objects are both investigated. The integration of range and intensity information is realized using range flow and optical flow constraints, which have been used for motion estimation in range and intensity images respectively. Range flow and optical flow lead to similar mathematical formulations, therefore they are well integrated into one estimation problem. Using these *flow* algorithms, first a method of estimating relative orientation of a pair of camera frames is presented. A highly over determined system of equations is solved for estimating the six parameters of relative orientation between two frames of the range camera. The trajectory of the moving camera is then computed using sequentially estimated pair wise relative orientations. The concatenation of sequential pair wise orientations lead to drift and accumulation of errors which does not give a globally consistent trajectory. To solve this problem, a method utilizing relative orientations results in bundle adjustment is presented. Matching distinctive features in images helps to identify loop closures and revisit of an area, which is essential in obtaining a globally consistent trajectory. However, in indoor environments features may be sparse and due to similar looking environment robust feature matching can be very challenging. Thus, the solution proposed in this thesis, utilizes the estimated relative orientations in the bundle adjustment. So, even when the feature points are low in number and not well distributed across the image, the orientation can still be accurately estimated by using information from the relative orientation. The proposed algorithm is evaluated on a publicly available dataset and benchmark, which shows that the algorithm performs well in comparison to the state of the art algorithm. Furthermore, using variance component analysis in bundle adjustment, it is shown that the original accuracy estimates of the relative orientation are far too optimistic.

Furthermore, this thesis presents a method for dense 3D motion estimation of independently moving objects with a static camera, which is also based on the integration of range flow and optical flow constraints. This method is based on two steps, in the first step the motion is estimated locally, while in the second step a global regularization is performed, which leads to

smooth dense flow vectors. The advantage of such an approach is that it leads to a linear equation system, which is then iteratively solved to remove the outliers.

In the end, an example of motion estimation on a landslide is presented. The motion estimation is realized using range flow constraint, which is applied on raster based digital surface models generated from the multi-temporal laser scanning data of a landslide surface.

The thesis demonstrates the feasibility and the benefits of integrating range and intensity data, of combining global and local models, and finally of considering stochastic properties of the measurements in the parameter estimation.

Kurzfassung

In den letzten Jahren haben 3D-Kameras mit hohen Bildwiederholfrequenzen und günstigem Anschaffungspreis eine große Verbreitung gefunden. Diese Kameras erlauben gleichzeitig Entfernung und Intensität zu messen und sind deshalb für viele Anwendungen interessant. Die Vermessung von Innenräumen und die selbsttätige Auto-Navigation seien als Beispiele genannt. Die Bestimmung der Bewegung der Kamera bzw. jener der beobachteten Objekte ist eine wichtige Teilaufgabe in all diesen möglichen Anwendungen. Deshalb ist es wichtig Methoden zu untersuchen, die die Bewegung unter Verwendung der simultan erfassten Strecken- und Intensitätsmessungen bestimmen.

Diese Arbeit untersucht diese gemeinsame Verwendung von Strecken- und Intensitätsmessungen für die Bewegungsbestimmung. Dabei wird die Bewegungsbestimmung sowohl von einer bewegten Kamera als auch von mehreren sich unabhängig bewegenden Objekten untersucht. Die gemeinsame Verarbeitung der Entfernungs- und Intensitätsmessungen wird über Range-Flow und Optical-Flow-Bedingungen realisiert. Der in Entfernungsbildern formulierte Range-Flow und der in Intensitätsbildern formulierte Optical-Flow verwenden sehr ähnliche mathematische Beschreibungen. Daher lassen sich beide sehr gut in einer gemeinsamen Parameterschätzung zusammenfassen.

Anhand dieser Flow-Algorithmen wird zuerst die relative Orientierung eines Paars von aufeinanderfolgenden Bildern bestimmt. Dabei wird ein hochgradig überbestimmtes Gleichungssystem gelöst um die sechs Parameter der relativen Orientierung zu berechnen. Die Trajektorie der bewegten Kamera ergibt sich dann aus der Sequenz aller paarweise berechneten relativen Orientierungen. Diese paarweise Aneinanderreihung führt zu einer Aufsummierung von Fehlern, was sich in einem Gangfehler in der berechneten Trajektorie niederschlägt. Um dieses Problem zu lösen und somit eine global konsistente Trajektorie zu bestimmen, werden die Ergebnisse aller relativen Orientierungen in einer gemeinsamen Bündelblockausgleichung eingeführt.

Für die Schätzung einer global konsistenten Trajektorie ist es wichtig denselben Objektbereich mehrmals zu erfassen und diese so entstehenden Schleifenschlüsse zu identifizieren. Für Letzteres werden eindeutige Merkmale in den Bildern extrahiert und gematcht. In Innenraumbereichen können diese Merkmale selten auftreten wenn die Umgebung entweder einfärbig ist oder über sich wiederholende Muster verfügt. In solchen Szenarien sind die Extraktion und das Matching von robusten Merkmalen somit sehr schwierig. In dieser Arbeit wird dieses Problem durch die geschätzten relativen Orientierungen in der Bündelblockausgleichung gelöst. Somit kann die Orientierung auch dann noch genau geschätzt werden, wenn die Anzahl der Merkmale gering oder deren Verteilung ungünstig ist.

Der vorgestellte Algorithmus wird anhand einer öffentlichen Datenbank an Bildern getestet. Die Einordnung der Performance des Algorithmus in die dort veröffentlichten Vergleichswerte („Benchmark“) zeigt, dass er sich im Vergleich zu anderen aktuellen Methoden sehr gut hält.

Die weitere Analyse anhand der Varianz-Komponenten-Schätzung in der Bündelblockausgleichung hat gezeigt, dass die original geschätzten Genauigkeitswerte der relativen Orientierung viel zu optimistisch ausfallen.

Im letzten Teil der Arbeit wird eine Methode vorgestellt, wie anhand einer statischen Kamera die Trajektorien von sich unabhängig voneinander bewegenden Objekten bestimmt werden kann. Auch hier wird wieder auf die gemeinsame Verwendung von Range-Flow und Optical-Flow zurückgegriffen. Die Bestimmung erfolgt in zwei Schritten. Im ersten Schritt werden lokale Bewegungsvektoren berechnet. Im zweiten Schritt werden diese einer globalen Optimierung unterzogen wodurch sich dann ein geglättetes dichtes Feld von Bewegungsvektoren ergibt. Der Vorteil dieses Zugangs liegt darin, dass es auf ein lineares Gleichungssystem führt, welches iterativ gelöst wird um grobe Fehler zu entfernen.

Den Abschluss der Arbeit bildet ein praktisches Beispiel, bei dem die Bewegungsvektoren aufgrund eines Erdrutsches aus multi-temporalen Laserdaten bestimmt werden. Die Berechnung verwendet den Range-Flow und wendet diesen auf rasterbasierte digitale Oberflächenmodelle an.

Zusammenfassend wurde in dieser Arbeit gezeigt, dass die gemeinsame Verwendung von Entfernungs- und Intensitätsmessungen, die Kombination von lokalen und globalen Modellen und die Berücksichtigung der stochastischen Eigenschaften der Messungen bei der Parameterschätzung durchführbar und auch von Vorteil sind.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Objectives

Motion analysis is an important topic in photogrammetry, computer vision, robotics and remote sensing, which involves motion of a sensor in a static environment and the motion of independently moving objects as well as motion relating to dynamics of the natural phenomena like landslides or glaciers. The knowledge or information of the environment can be acquired by a sensor like a camera mounted on a platform moving through the surroundings or a camera used hand held. Subsequently, estimating motion of the camera is an essential task in utilization and processing of the acquired data. For autonomous navigation as well as for scene reconstruction its not only important to determine the motion of the platform itself but it may also be of interest to determine motion of independently moving objects in the surroundings.

In the recent years low cost high frame rate 3D cameras have become widely available. Commercial availability of these low cost sensors along with ease of use has derived interest in numerous applications. The goal of this thesis is motion estimation in data from range imaging sensors like Time of flight (ToF) cameras or structured light based cameras. These type of cameras are sometime called as RGBD or RGBZ sensors as they can provide RGB color image along with depth of each pixel. The emphasis in this thesis is on determining motion of the camera and motion of independently moving objects. The motion of the camera is defined by its exterior orientation consisting of six parameters: three parameters defining the position of the camera perspective center and three parameters for defining the angular attitude of the camera with reference to a superior coordinate system. In case of motion of a camera, terms like camera pose estimation, camera trajectory, ego-motion estimation and visual odometry all refer to determination of exterior orientation parameters of a moving camera. The range sensors typically also provide intensity information along with the distance information. While a lot of work has been done on determining the orientation of the camera from intensity images, the optimal fusion of intensity and range information for determining camera orientation still requires further investigations. The system investigated is based on a single camera, where the motion is generated by a moving camera or independently moving objects. Furthermore,

experimental data consists of image sequences or video, which implies the temporal ordering of images and presumes small motion and high overlap of the scene in consecutive frames.

The technology of range imaging sensors continues to develop, however there are still limitations in the spatial resolution, accuracy of distance measurement, background light subtraction and maximum measurable distance. Due to these limitations range imaging cameras are mostly suitable for indoor close range applications. Therefore, mapping and monitoring of the indoor environment is a key application for these type of cameras. Indoor environment is usually composed of planar objects like walls and ceilings and often image texture on these surfaces is very low. The state of the art image based techniques for determining the camera orientation make use of distinctive image features. In an indoor environment the number of distinctive features may be low and these features may not be well distributed throughout the image, which will have an adverse effect on the accuracy of estimated camera orientation. Similarly, the accuracy of methods for point cloud or range image registration will be effected by limited 3D structure or geometry. Consequently, the complementary intensity and range information as available in modern day 3D sensors is a valuable asset as it can help to achieve better results in such applications. Therefore, it is essential to investigate methods which perform optimal integration of range and intensity information for the task of motion estimation.

The research problem investigated in this thesis is the optimal fusion of the range and intensity information for the task of estimating motion of camera and motion of independently moving objects with a static camera. An important aspect of estimating camera motion is to obtain a globally consistent solution of the camera trajectory. On the other hand, while studying motion of independently moving objects, it is desired to estimate 3D motion vector for each pixel of the image, which is subsequently useful in segmenting independently moving objects. Furthermore, it is desired that the derived methods should consider proper stochastic modeling of the observations, because by stochastic modeling, an estimate of the accuracy of the solution can be obtained. Furthermore, if the stochastic model is correct, then the least squares is the best linear unbiased estimator.

1.2 Overview

In this thesis the integration of range and intensity data is realized using optical flow and range flow constraints, which lead to a similar formulation in terms of motion parameters. As a result, a joint estimation problem is solved which combines information from both channels in a single framework. Both camera motion and motion of independently moving object can be investigated using this strategy. Range flow and optical flow algorithms typically estimate motion between a pair of images. If more views or images are available then the pairwise motion is sequentially computed. Using pairwise transformation does not give a globally optimal estimate due to accumulation of errors over longer sequences. A solution based on integration of relative orientation into bundle adjustment is proposed to obtain globally optimal estimate of camera trajectory. Bundle adjustment is the state of the art method for obtaining a globally optimal solution of camera poses and 3D structure. Therefore, complementing bundle adjustment with relative orientation is expected to produce better results especially in scenarios with limited scene texture which is often the case in indoor environments.

The motion estimation of independently moving objects is also based on integration of range flow and optical flow constraints. The task of dense motion estimation is realized using a two step procedure. In the first step, information from the local neighborhood is used for estimating motion at each pixel and in the second step a global regularization is performed over the whole image to compute smooth dense motion vectors at each pixel with sharper motion boundaries.

The layout of the thesis is as follows: In Chapter 2 theory and the corresponding related work in area of motion estimation using range and intensity data is presented, this includes topics of multiview geometry, point cloud registration, flow algorithms and fusion of range and intensity data. The inclusion of all these topics is necessary due to the combination of data and methods utilized in this work for the task of motion estimation. In Chapter 3 a brief introduction of ToF cameras, active triangulation sensors and laser scanners is given as data from these sensors is used for evaluation of the methodology. The method of relative orientation using direct methods and the incorporation of relative orientation in bundle adjustment is presented in Chapter 4. Then the method of motion estimation of independently moving objects is presented in Chapter 5. A brief discussion on estimation models used for flow algorithms is included in Chapter 5. The evaluation of the proposed methods is presented in Chapter 6. Both qualitative and quantitative evaluation of the methods is presented. Evaluation of relative orientation and bundle adjustment with relative orientation constraints is performed on publicly available RGB-D dataset [37]. Sequences from ToF camera are also used for the evaluation of the proposed methods. Motion estimation has several applications in area of remote sensing. In Chapter 6 an application of range flow to estimate motion of a landslide is presented. Finally, conclusions along with the future prospects are given in Chapter 7.

1.3 Contributions

The first contribution of this work is to embed the developments in optical flow and range flow algorithms in to a framework for direct estimation of camera relative orientation or ego motion. These developments include coarse to fine warping strategy and robust estimation. Although the application of direct methods for motion estimation using intensity and range data has already been presented in [67, 76, 78], this work extends these methods to robustly estimate relatively large motions (Chapter 4).

The second main contribution is the integration of relative orientation constraints from direct methods into bundle adjustment (Chapter 4). This method combines the advantages of sparse feature matching and dense image and point cloud registration to estimate globally consistent solution of camera motion. More specifically, relative orientation constraints along with their estimated covariance matrices are introduced as observations in the bundle adjustment, which simultaneously optimizes camera orientations and the 3D point locations.

The third contribution is a global regularization procedure for estimating dense 3D motion vectors of independently moving objects in an ordinary least squares framework (Chapter 5). This method extends the work of Spies et al. [177], to include an-isotropic smoothing and robust estimation in global regularization of the 3D motion vectors. This global regularization procedure solves a linear system of equations using ordinary least squares to obtain dense smooth motion vectors.

The fourth contribution is the application of range flow over a landslide to estimate the 3D displacements. In the discipline of geomorphology, understanding the dynamics of a landslides is an important task and motion estimation of the landslide is essential part of this task. The surface of a dynamic landslide exhibits a complex motion pattern, which is analyzed by applying the range flow constraint on surface models of the landslide generated from laser scanning data at different instants in time.

Furthermore, the stochastics of the observations are take into account in the motion estimation model and a discussion on least squares model for motion estimation is presented (Chapter 5). The integration of range flow and optical flow is realized using a least squares solution where the stochastic properties of each distance and intensity measurement can be taken into account. Furthermore, the results of relative orientation and local motion estimation are used along with the variances of the estimated parameters, which then performs the weighting of the terms or observations in the next estimation step.

The papers published during the course of this research are [49–52, 96, 97, 144, 157]. In [49] and [50] the method of estimating motion and segmentation of multiple moving objects using integration of range flow and optical flow is presented. Chapter 5, contains the method presented in these papers. The final aim of this work was the segmentation of independently moving objects in range video sequences. The motion vectors derived using the method presented in these papers were used to generate trajectories for each pixel of the image and a graph based segmentation algorithm was used to segment these trajectories into independent moving objects [49, 50]. The work related to segmentation of trajectories was done by co-author Nicole Brosch [16, 17, 49, 50] as part of joint research collaboration in context of *Doctoral College on Computational Perception*.

Estimation of camera relative orientation using direct methods based on integration of range and intensity data is presented in [51]. In this work it was shown that due to robust estimation it is possible to filter out the independently moving objects from the static parts of the scene if the dominant motion is due to camera motion.

The quantification and modeling of the scattering errors in ToF camera is investigated in [96,97]. Quantification of this error and calibration and modeling of systematic errors is essential part of data acquisition and analysis of data from the range sensors. This work was performed in support of the Tof camera calibration research done by Wilfried Karel [94, 95]. In [97] the scattering was modeled using a point spread function and a deconvolution procedure was applied to estimate the image without scattering distortion.

In [52] and [157] the motion estimation of the landslide using range flow is investigated and compared with the results of point based geodetic measurements. Roncat et al. [157] further includes a comparison between surface models generated from airborne laser scanning data and image matching using an unmanned air vehicle.

The work in [144, 198] was part of a research work to develop automatic classification tools for classification of airborne laser scanning data. The geometrical and radiometric features derived for each laser scanning point were used for classifying points into various land cover types. In [144] the focus of the paper was on the features derived for each point and the management of the point cloud data.

CHAPTER 2

Theory and Related Work

In this chapter the theoretical background and state of the art methods for motion estimation with focus on the orientation of the moving camera and motion of independently moving objects in data from range sensors is presented. Traditionally there has been a lot of studies done in motion estimation from images but as the range sensors have become widely available in the recent years, motion from range data has also become very popular [27]. As both intensity and range information is available in the range sensors, methods developed for both intensity and range images are relevant in the scope of this work. Therefore, the theory and related work can be divided into methods developed for only intensity data or range data or methods based on fusion of range and intensity data. Furthermore, motion estimation using flow algorithms is a rather vast topic in itself, therefore, the related work has been further divided to discuss flow algorithms separately as they form a core component of the methodology proposed in this thesis. The topics discussed here have been developed and applied in photogrammetry, computer vision and robotics communities, therefore, the body of literature corresponding to these topics is vast. However, some of the methods are principally very similar but named differently due to their development in different disciplines.

2.1 Camera Orientation and Multiple View Geometry

A camera projection model defines the mapping of the 3D object space to the 2D image space. Throughout this work a pinhole camera model is assumed. Therefore, the distortion is either deemed to be negligible because the random errors are by far larger than the systematic errors or it is assumed to be constant and determined beforehand using camera calibration and subsequently removed [107, 126]. The imaging geometry is shown in Figure 2.1. All rays from the scene points intersect at a common point known as camera projection center O. In Figure 2.1 the the coordinates of the projection center are (X_0, Y_0, Z_0) in the global coordinate system (X, Y, Z). The distance from the projection center to the image plane gives the principal dis-

tance f [1]. The image coordinate of a perpendicular ray from projection center to the image plane gives the principal point (x_0, y_0) in the camera coordinate system (x, y, z).

The derivation of the relation between the image and object coordinates given here, closely follows [108], where a detailed derivation is available. Suppose a coordinate system (X', Y', Z') parallel to (x, y, z). Due to collinearity of the points $OP'P$ (Figure 2.1), the following relation can be written.

$$\begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X' - X'_0 \\ Y' - Y'_0 \\ Z' - Z'_0 \end{bmatrix} \tag{2.1}$$

where r_{ij} are the elements of the rotation matrix, which rotates the coordinate axes of X, Y, Z to align with X', Y', Z'. Further discussion on the rotation matrix is given later in the thesis (Chapter 4). The relations between the object coordinates and image coordinates of point P' are give as:

$$\begin{aligned} \frac{x - x_0}{f} &= \frac{X' - X'_0}{Z' - Z'_0} \\ \frac{y - y_0}{f} &= \frac{Y' - Y'_0}{Z' - Z'_0} \end{aligned} \tag{2.2}$$

Using the relations in Eqs. (2.2) and (2.1) the so called collinearity equations are written as:

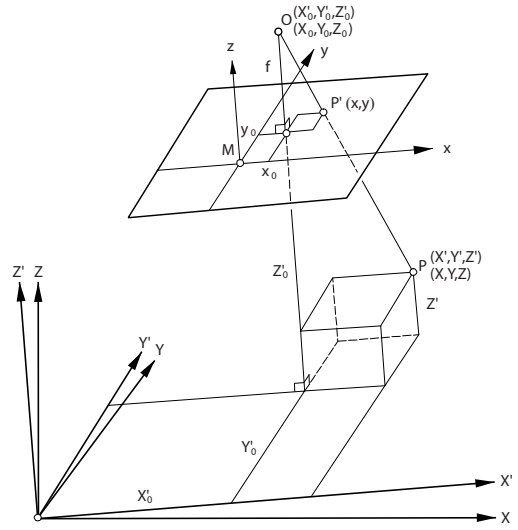

Figure 2.1: Geometry of Image formation. (adapted from [108]). The coordinate system (X', Y', Z') is parallel to image coordinate system (x, y, z) which is rotated with respect to coordinate system (X, Y, Z).

[1] In Photogrammetry, principal distance is usually denoted as c. However, in this thesis c is used to denote speed of light, therefore, f is used here for principal distance.

$$
\begin{aligned}
x &= x_0 - f\frac{r_{11}\left(X - X_0\right) + r_{21}\left(Y - Y_0\right) + r_{31}\left(Z - Z_0\right)}{r_{13}\left(X - X_0\right) + r_{23}\left(Y - Y_0\right) + r_{33}\left(Z - Z_0\right)} \\
y &= y_0 - f\frac{r_{12}\left(X - X_0\right) + r_{22}\left(Y - Y_0\right) + r_{32}\left(Z - Z_0\right)}{r_{13}\left(X - X_0\right) + r_{23}\left(Y - Y_0\right) + r_{33}\left(Z - Z_0\right)}
\end{aligned}
\tag{2.3}
$$

The collinearity equations represent the relation between camera exterior orientation (the position of the projection center (X_0, Y_0, Z_0) and the angular attitude of the camera), interior orientation (principal point (x_0, y_0) and principal distance f) and image observations which are e.g. image coordinates of feature points. The exterior orientation of the image can be computed indirectly from control points using the method of space resection. If the interior orientation of the image is known then a minimum of three points (with known coordinates in (X, Y, Z); the so called *control points*) are required to compute the orientation of the image using collinearity equations (Eq. (2.3)). As collinearity equations are non linear, an approximate solution should be provided for estimating orientation of the image using this method of space resection. Using the method Direct Linear Transformation (DLT) [1], orientation of the image can be estimated by solving a linear system of equations without any need of approximate solution [118]. The direct orientation of the image can be computed using e.g. GPS and an inertial measurement unit (IMU) [28, 68, 69, 209].

The range is the distance from the projection center to the object point, which according to Figure 2.1 can be written as:

$$
D_R = \sqrt{(X' - X'_0)^2 + (Y' - Y'_0)^2 + (Z' - Z'_0)^2}, \tag{2.4}
$$

In this thesis the *range* refers to the distance D_R, while depth is the $Z' - Z'_0$ coordinate of the object point. The range flow constraint used in thesis is based on the $Z' - Z'_0$ coordinate observation of each pixel. The range D_R is used in the bundle adjustment as the distance measurement to the object point.

Now, if multiple images from different view points are available (which in this thesis are assumed to be generated by motion of camera), techniques of multi-view geometry can be applied to determine relative orientations of these camera positions. Camera relative orientation from two or more views has been extensively studied in photogrammetry and computer vision communities. Figure 2.2, shows the geometry of relative orientation from two views. Given a corresponding point between two images, the so called *coplanarity constraint* means that the object point, the corresponding image points in both images and the projection centers all lie on a single plane [108, 118, 128], which can be written as:

$$
\mathbf{b}^T \left(\mathbf{p}_1^{'} \times \mathbf{p}_2^{'}\right) = \begin{vmatrix} b_x & p_{1,X} & p_{2,X} \\ b_y & p_{1,Y} & p_{2,Y} \\ b_z & p_{1,Z} & p_{2,Z} \end{vmatrix} = 0 \tag{2.5}
$$

Here, $\mathbf{b}$ is the baseline vector in the coordinate system (X, Y, Z) (also known as model coordinate system), $\mathbf{p}_1^{'}$ and $\mathbf{p}_2^{'}$ are the vectors to the point $\mathbf{P}$. The coordinate system as shown in Figure 2.2 was chosen so that the baseline between two camera positions is along the X-axis.

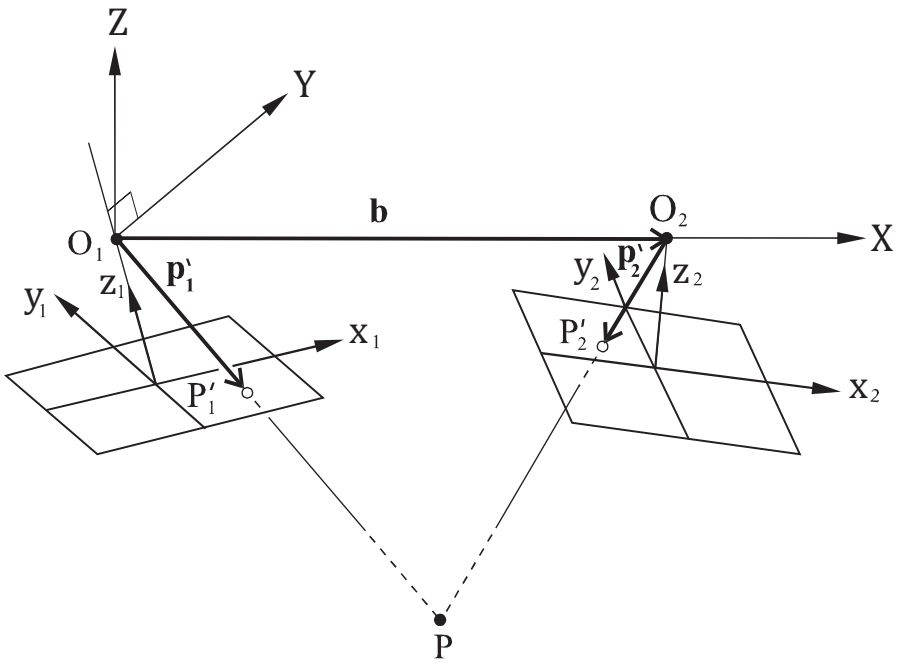

Figure 2.2: Relative orientation of two images. (adapted from [108]). The baseline between two images is along the X axis of the model coordinate system XYZ

Using the coplanarity constraint, the transformation between two images can be written in form of essential matrix and fundamental matrix for calibrated cameras and uncalibrated cameras respectively [66, 186]. Several algorithms exists for computing the essential and fundamental matrices from corresponding points in the two images [66, 126, 140]. The translation and rotation components are then computed from the essential or fundamental matrices. Details of these procedures can be in found in relevant texts [66, 126]. If false corresponding point matches exist then techniques like RANSAC [41] or M-estimators [84] are used to find a set of inlier points. Similarly in the case of three views the relative orientations of the three images may be encoded in the trifocal tensor [66, 73, 151]. These relative transformations are, however, defined up to a scale factor which means that multiplying them with a scalar value still satisfies the coplanarity constraint. The position of the corresponding points in 3D space can be computed by triangulation of the image rays from estimated relative orientations. Figure 2.2 shows the geometry of relative orientation between the two images, if the absolute orientation of the image pair is desired, then information through control points or GPS should be included.

When relatively large sets of images with different viewpoints are given, the task of determining camera orientations and sparse 3D structure of the scene is known as structure from motion [119, 184, 193]. Bundle block adjustment [18, 57, 108, 118, 126, 128] gives the most accurate solution of the structure from motion problem. Here *bundle* refers to bundle of rays from 3D points towards camera perspective center and the bundle block adjustment means the estimation of position and orientation of bundle of rays [128]. Compared to relative orientation

using two or three images, bundle adjustment can simultaneously optimize large number of camera orientations, point coordinates and camera calibration parameters [190]. It is a non linear optimization, which refines the initial estimates of the structure i.e. coordinates of the object points and camera parameters. The basis for bundle adjustment are the collinearity equations (Eq. (2.3)) which give the relationship between the observed image coordinates, 3D position of points and camera interior and exterior orientations. Bundle adjustment is a flexible technique as different types of observations, like control points, distance observations and constraints can be included in the estimation procedure. Bundle adjustment is a non linear optimization due to the non linearity of the observation equations, therefore an approximate solution is required for initialization of the bundle block adjustment. Relative orientation methods and factorization methods [66, 181, 189] are commonly used for initial approximations of structure from motion problem [66]. Structure from motion has been used for many years for mapping and 3D reconstruction applications [147, 172].

In mobile robotics simultaneous localization and mapping (SLAM) algorithms are used for tracking robot pose or orientation together with mapping of the robot environment. The goal is that the robot should be able to autonomously navigate in an unknown environment with an unknown initial position. In SLAM algorithms there is a high emphasis on fast or real time computation because the map of the landmarks (e.g. feature points) and the position of the robot needs to be computed online so that the robot can autonomously navigate and avoid obstacles in an unknown environment. SLAM methods often use information from inertial measurement unit and other sensors like wheel encoders on board a moving robot. In addition to visual sensing, robots may also be equipped with laser-range finder or sonar based sensors. In context of this thesis, however, the main interest lies in the methods that solely rely on camera based navigation, such algorithms can be categorized as *visual SLAM* algorithms. In visual SLAM algorithms the task of estimating robot pose is in fact the problem of determining exterior orientation of a moving camera which is the topic discussed in this thesis. The monocular SLAM method [31, 32], utilizes only visual information from the camera for computing camera trajectory and mapping of the environment. This method uses matching of image patches between frames to compute motion in an extended Kalman filtering framework. Many of the SLAM algorithms use Kalman filtering for estimating the next camera pose and updating the poses at the next time interval.

The techniques for visual SLAM and structure from motion algorithms are principally similar as both of these methods try to estimate both camera orientations and 3D structure. Bundle adjustment for large number of images can be computationally expensive, therefore, bundle adjustment is often applied to a subset of recent images for online SLAM [40, 101]. In structure from motion, however, the main interest lies in the visual reconstruction and not on the orientation parameters of the camera. Furthermore, most of the structure from motion algorithms run off line, while in SLAM the emphasis is on online estimation of the positions and the map. However, this difference is not always true, as some structure from motion algorithms are capable of running online [141] and often the optimization of SLAM poses is done offline [187]. For large scale mapping problems, the number of map features and robot poses become enormous and it becomes increasingly difficult to optimize such a large equation system. Therefore, SLAM algorithms often use only the robot poses as variables and optimize over the robot poses

and pairwise constraints to achieve globally consistent solutions. Many of the SLAM algorithms [58, 59, 104, 116, 143] have build upon the idea of Lu and Milios [116], who used the robot poses as nodes of the network or graph, pairwise transformation and constraints as links between the nodes and performed a least squares optimization over the this network to obtain a global solution (also known as *pose graph optimization*). The constraints arise from revisit of certain areas or loop closures which can be identified using feature matching. The sparsity in this type of network or graph can be used for solving large number of poses and constraints. In the recent years the emphasis of the graph based SLAM algorithms has been on fast convergence and solution of large number of nodes [59, 104, 105, 143, 187]. The pose graph estimation problem in SLAM is principally the same problem as geodetic network adjustment studied in geodesy and surveying since long time [4]. In a Geodetic network the observations are the surveyed angular and distance measurements between station points. Both SLAM and geodetic network adjustment [71] deal with large scale problems with non linear constraints among the nodes. For example, large scale SLAM problems optimize upto 100,000 robot poses [104] while the readjustment of North American geodetic network required solving for positions of 200,000 stations [53, 103].

The methods of orientation described here makes use of corresponding feature points in the images. The procedure of automatically finding corresponding points among different images comprises of feature detection, feature description, feature matching and optionally a fine measurement of the feature point coordinate using area based matching [108] as is often done in aerotriangulation [108, 126]. These distinctive feature points should be ideally invariant under scale, rotation and illumination changes, so that they can be robustly and repeatedly matched in images with different viewpoints. Feature detectors or interest point operators detect points or region in images with sufficient gradient information, which can be matched in different images. Corner detectors [43, 65] and blob detectors [123, 129] are commonly employed interest point operators. The information of each feature point and its neighborhood is stored in the descriptors like SIFT descriptor [115], GLOH [130] and SURF descriptor [11]. The feature matching is done by computing distances between the descriptors in the descriptor space and finding the best match e.g. the nearest neighbor which is below a threshold [115, 130]. A detailed survey of state of the art feature detectors, their evolution and pros and cons is presented in Tuytelaars and Mikolajczyk [192].

State of the art algorithms in visual SLAM and structure from motion are based on sparse feature point matching and bundle adjustment. Therefore, these method will have limitations in case when feature points are low in number and not well distributed over the image. Hence, it becomes essential to utilize range information for more accurate estimation of motion. The method proposed in this thesis complements feature matching with dense pixel wise matching to achieve better results. As compared to pose graph estimation methods of SLAM, the method presented in this thesis performs a bundle adjustment, which simultaneously optimizes the 3D structure and camera poses using the pairwise constraints in the adjustment.

2.2 Point Cloud Registration

Given two sets of 3D points, the goal of registration is to find the optimal transformation between the two data-sets. The transformation is optimal in the sense that it minimizes some sort of distance between the corresponding points. These set of points represent the surfaces of the objects in the scene, and the optimal transformation aligns these two surfaces. So, given two point sets $\{\mathbf{d}_i, \mathbf{m}_i; i = 1, 2, ..., N\}$, the goal is to find the rotation $\mathbf{R}$ and translation $\mathbf{T}$ which minimizes the distance between the corresponding points:

$$E = \sum_{i=1}^{N} \left(\mathbf{R}\mathbf{d}_i + \mathbf{T} - \mathbf{m}_i\right)^2 \tag{2.6}$$

here, $\mathbf{d}_i = (d_{ix}, d_{iy}, d_{iz})^T$ and $\mathbf{m}_i = (m_{ix}, m_{iy}, m_{iz})^T$ are i^{th} corresponding points of $\{\mathbf{d}_i\}$ and $\{\mathbf{m}_i\}$. In above formulation a seven parameter similarity transformation can also be used which also includes the scale factor between the two point sets. Generally, the point correspondences between the two data-sets is not known. The Iterative Closest Point (ICP) [12,24,211] algorithm iteratively establishes the correspondences between the two point sets by using the closest point as the corresponding point to minimize Eq. (2.6). ICP is the state of the art method used for registration of two point data sets. ICP algorithm was independently proposed in [12, 24, 211]. Besl and Mckay [12], minimized the point to point distance as in Eq. (2.6), while Chen and Medioni [24] minimized the point to plane distance as given by:

$$E = \sum_{i=1}^{N} \left[\left(\mathbf{R}\mathbf{d}_i + \mathbf{T} - \mathbf{m}_i\right) \cdot \mathbf{n}_i\right]^2 \tag{2.7}$$

here $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz})^T$ is the unit normal vector at point $\mathbf{m}_i$. Variants of ICP with different formulations have been proposed. A good comparison can be found in Rusinkiewicz and Levoy [159]. ICP algorithms require good approximate registration of the two point sets otherwise the solution can get stuck into local minima. Often a threshold is applied to the distance of closest point to remove the outliers from the corresponding points. The minimum of the point to point distance can be computed using the closed form solution based on singular value decomposition [54, 179](SVD) [7], orthonormal matrices [82], quaternions [79] and dual quaternion [199]. Eggert et al. [36] has compared these four closed form solutions and found no clear differences in stability and accuracy of these methods. The minimization of point to plane distance metric is done iteratively in the form of linear least squares estimation. The minimization of Eq. (2.6) using the SVD [54,179] is briefly described here as it is used for finding approximate orientation of image pairs later in thesis. Let the centroid of two points sets $\{\mathbf{d}_i\}$ and $\{\mathbf{m}_i\}$ be $\overline{\mathbf{d}}$ and $\overline{\mathbf{m}}$ respectively. The coordinates of the points with reference to the centroid are:

$$\overline{\mathbf{d}}_i = \mathbf{d}_i - \overline{\mathbf{d}} \qquad \overline{\mathbf{m}}_i = \mathbf{m}_i - \overline{\mathbf{m}} \tag{2.8}$$

The points $\{\mathbf{R}\mathbf{d}_i\}$ and $\{\mathbf{m}_i\}$, will have the same centroid [83]. Therefore, Eq. (2.6) can be rewritten as

$$E = \sum_{i=1}^{N} \left(\mathbf{R}\overline{\mathbf{d}}_i - \overline{\mathbf{m}}_i\right)^2 \tag{2.9}$$

Now rotation and translation can be estimated separately. The minimization of Eq. (2.9) is achieved when $Trace(\mathbf{RH})$ is maximized, where

$$\mathbf{H} = \sum_{i=1}^{N} \overline{\mathbf{d}}_i \overline{\mathbf{m}}_i^T \tag{2.10}$$

If the SVD of $\mathbf{H}$ is $\mathbf{USV}^T$, then the optimal $\mathbf{R}$ which maximizes $Trace(\mathbf{RH})$ is given as:

$$\mathbf{R} = \mathbf{VU}^T \tag{2.11}$$

The translation $\mathbf{T}$ is then computed as:

$$\mathbf{T} = \mathbf{R}\overline{\mathbf{d}} - \overline{\mathbf{m}} \tag{2.12}$$

In case that the points are coplanar, the determinant of $\mathbf{R}$ can be -1, which indicates a reflection instead of a rotation. In such degenerate cases $\mathbf{R}$ can be computed as $\mathbf{R} = \mathbf{V}'\mathbf{U}^T$, where $\mathbf{V}'$ is computed from $\mathbf{V}$ by changing the sign of the column of $\mathbf{V}$ corresponding to the zero singular value of $\mathbf{H}$ [7, 36]. Using Eqs. (2.11) and (2.12), the rotational and translational parameters corresponding to the least squares fitting of two point sets with known correspondences can be computed. In case of image feature matching with known distance to the object point (as is available in range sensors), this method can be used to find the transformation between the two images.

When multiple point clouds are available, registration is typically done sequentially using pairs of point clouds. This results in accumulation of errors and drifts which does not yield a globally optimal solution. Several techniques have been proposed to find more globally optimal solution for multiple point clouds. In Chen and Medioni [24] ICP paper, a global solution for registration of multiple scans based on metaview was given. In this approach each point cloud was registered to previously registered point clouds, i.e a point cloud formed by registration of scans was sequentially generated and it was called metaview and the successive neighboring point clouds were registered to this metaview. Lu and Milios [116] proposed a network based solution of registering multiple scans by solving all poses as variables and introducing pairwise transformation as observations. Sharp et al. [167] have also proposed a similar solution by optimizing a graph of spatial relationships of neighboring views. Ressl et al. [153] have performed simultaneous least squares adjustment of airborne laser scanning strips using the pairwise strips transformation derived using LSM [152, 153]. Pulli [148] has proposed to incrementally align each view to a set of consistently aligned views by starting from the view with the most number of connections. A recently proposed method named kinectfusion [86, 137] also integrates the depth data into a global surface model and the registration of each camera pose is done by ICP alignment of surface models with the current depth image of Kinect data.

Least Squares Matching (LSM) [5, 63, 110, 153] which is principally similar to ICP has also been used for registration of surfaces. Methods utilizing the distribution of surface normals over a surface area, such as spin images [88] and extended Gaussian images [81] have been used for surface matching. These methods can be used for finding approximate solutions for initialization of the ICP algorithm which then determines fine transformation parameters.

The point cloud and surface registration methods like ICP which use geometric information will not be able to uniquely determine all the transformation parameters if the surface under consideration is e.g. a plane or a cylinder. Therefore, it is vital to utilize the intensity information as well which may help to uniquely and robustly determine all the transformation parameters [205]. Furthermore, for the case of registering multiple point clouds several methods have been proposed but there is no consensus on the standard approach to apply for registering multiple point clouds. The method presented in this thesis (Chapter 4), addresses both these issues as it simultaneously uses range and intensity information and makes use of bundle adjustment for computing a globally optimal solution which is a standard in multi-view image registration.

2.3 Flow Algorithms

In this section, the *flow* algorithms: *optical flow* and *range flow* are described. In contrast to sparse feature based approaches these method use dense pixel information for estimating motion. These methods have been used for estimating motion of camera as well as the motion of the independently moving objects with the larger body of literature concentrated on the latter problem as the problem of estimating dense motion with sharp boundaries is a difficult task with many applications requiring fast, accurate and robust motion estimation e.g self driving cars. Techniques utilizing flow constraints for estimating camera motion parameters are known as *direct methods* [76, 78] as they determine the unknown parameters directly from the measured image quantities such as intensity and depth without computing features or explicitly computing flow for each pixel [85].

Optical Flow

Optical flow is the problem of estimating 2D image motion in image sequences. The task is to find corresponding pixels between a pair of images, similar to finding corresponding pixels in a stereo image pair. The input for optical flow algorithms is typically an ordered image sequence or video. This means that motion between two consecutive images is typically small and temporal information can be utilized for enhancing the performance of the algorithms. Horn and Schunck [80] have pointed out the difference between *optical flow* and *motion fields*. Optical flow is the image motion which transforms one image of the sequence to the next image [80]. While, a motion field as defined by them is a purely geometric concept which is the projection of the 3D motion onto the 2D image and therefore has no ambiguity. Considering e.g. a simple example of an image containing only a line over a homogeneous intensity patch. In such an example only the motion perpendicular to the line can be estimated from the image and the motion along the direction of the line cannot be estimated unless there is more information available in the image. Thus, in this scenario optical flow will give motion perpendicular to the direction of the line. The goal is usually to estimate optical flow as close as possible to the motion field but this depends on how much information is available in the image [75, 80].

Most of the optical flow problems are based on brightness constancy, which assumes that the brightness or intensity of the pixel remains constant while moving from one image position to another in the next frame. Mathematically, the brightness constancy assumption can be written

as [9]

$$I(x, y, t) = I(x + \dot{x}, y + \dot{y}, t + 1), \tag{2.13}$$

here, $\dot{x}$ and $\dot{y}$ are the image motion and $t + 1$ corresponds to next instant in time. Eq. (2.13) is also known as brightness constancy constraint equation (BCCE) [162]. The function I is typically a nonlinear function corresponding to image brightness. The Taylor approximation of BCCE gives:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x}\dot{x} + \frac{\partial I}{\partial y}\dot{y} + \frac{\partial I}{\partial t} + ..., \tag{2.14}$$

neglecting the higher order terms, the well known optical flow constraint is obtained [77]:

$$I_x\dot{x} + I_y\dot{y} = -I_t. \tag{2.15}$$

Here I_x, I_y, I_t are the spatial and temporal derivatives of image brightness respectively and $\{\dot{x}, \dot{y}\}$ is the image motion or optical flow. The spatio-temporal derivatives are computed using derivative filters in the image space, which are discussed in detail in Chapter 4. Optical flow techniques based on Eq. (2.15) are known as differential optical flow techniques because they contain the spatio-temporal derivatives of the image brightness [10]. Differential techniques are most commonly used for solving optical flow problems [22]. Other techniques used for optical flow are e.g. region based [170] and phase based [201] techniques. In addition to brightness constancy assumption, state of the art optical flow algorithms also use gradient constancy assumption [19], normalized cross correlation and census for more robust performance [196].

Intensity of a single pixel is not enough to uniquely define the 2D motion of the pixel. Eq. (2.15) provides one constraint for each pixel with two unknown velocity components. Therefore, the optical flow is under-constrained and more information is needed to uniquely determine optical flow. This also leads to the so called aperture problem [121]. As mentioned earlier, if an edge is visible in the image, only the velocity component normal to the edge direction can be estimated while the component along the normal direction is ambiguous. The common way of solving this under-constrained problem is to embed information from the neighborhood by using smoothness prior. Differential optical flow techniques can broadly be divided into local and global methods based on the way this neighborhood information is exploited [22]. Local methods estimate flow using a window based neighborhood while assuming consistent flow in this local neighborhood and performing a least squares adjustment over an overdetermined systems of equations (Eq. (2.15)). A well known local optical flow method is Lucas Kanade optical flow [117]. Global methods estimate optical flow over the entire image by optimizing an energy function constituting the brightness constancy and smoothness constraint for each pixel. The seminal paper of Horn and Schunck [77] is an example of global optical flow method. Horn and Schunk [77] were the first to pose the optical flow problem a in variational framework [80] i.e using calculus of variation. The minimization of such a energy function is done by solving the corresponding Euler Lagrange equations [21]. In presence of strong texture, the brightness constancy constraint implicitly gets a higher weight and when there is no texture the smoothness term gets a higher weight.

The classical optical flow methods [77, 117] used quadratic penalizer, which is sensitive to the outliers. Black and Anandan [13] used robust penalty functions for mitigating violations of

brightness constancy and smoothness constraints. Zach et al. [210] used a L1 norm [158] for estimating optical flow. In presence of motion discontinuity, the smoothness constraint needs to be relaxed and large variations in image intensity are often cues for object boundaries. Therefore, an anisotropic smoothness term is often used for relaxing smoothness constraint [135, 206] across large variations in image intensity.

Most optical flow algorithms use first order approximation of brightness constancy. The image intensity is typically a nonlinear function. Therefore, after linearization only small motions (few pixels) can typically be estimated. For estimating larger motions, a well established solution is to use a coarse to fine warping strategy. At coarser level, motion is expected to be small. Therefore, first approximation of the motion is done at the coarse level and the next finer resolution image is warped according to these flow vectors. This step is repeated until the finest resolution and the optical flow is the aggregated motion from coarse to fine pyramid levels [19]. The coarse to fine strategy fails for small scale structure whose motion is very different from motion of larger scale structures. For example in the case of human body, motion of the arms can be very fast and different from the rest of the body. Brox and Malik [20] propose a solution based on feature matching with the variational framework to estimate large displacement optical flow. The introduction of feature matching into dense optical flow estimation is getting more attention in the recent work [25, 207]. Sun et al. [182] argue that bigger gains in the accuracy of optical flow algorithms will be achieved by incorporating reasoning about surfaces and boundaries and their motion over time. Recent top performing algorithms indeed include segmentation reasoning for separating individual surfaces and defining the boundaries [25, 183].

There have been several attempts for benchmarking optical flow algorithms [9, 10, 125, 145]. The quantitative comparison of Barron et al. [10] received great interest, however the comparison was limited in terms of the complexity of the synthetic images. Middlebury optical flow database [9] overcomes limitations of the previous benchmarks and has been used as a standard for comparison and evaluation of optical flow algorithms. However, the recent KITTI vision benchmark [48] offers more realistic and challenging optical flow sequences with ground truth.

In photogrammetry, the method of least squares image matching [2, 42, 60–62] has been used to compute transformation of pixels between two images. One popular application of this method is the generation of digital surface models from aerial imagery. The matching is realized by minimizing the squared sum of intensity differences in a window or an image patch, this is principally the same as the Lucas Kanade [117] optical flow. The process of aerotriangulation and subsequently the generation of digital surface model also makes use of image pyramids like the coarse to fine resolution in optical flow. In addition to minimization of intensity differences, geometric constraints like collinearity equations (Eq. 2.3) have also been introduced in the least squares adjustment. In comparison to optical flow, image matching applications often use images from larger baselines, and therefore, due to non linearity of the image intensity function an approximate solution is required. The unknowns in image matching are often the parameters of an affine transformation as by choosing a small image patch, the mapping between the image patches can be approximated by an affine transformation. However, image matching is a bit simpler problem in the sense that the epipolar geometry is computed from aerotriangulation, while generic optical flow can contain any arbitrary motion.

In optical flow only 2D image motion is estimated. The corresponding 3D motion estimation

problem from color or gray scale images is known as scene flow estimation [195]. As the depth of the image point is not known from a single image, scene flow problems often use multiple cameras. The task is then to perform a joint scene reconstruction using stereo matching [15] and full 3D motion estimation of each pixel [184, 203]. However, in this thesis the 3D information is given by the range sensors so multiple cameras are not required and furthermore, only the motion estimation problem is investigated.

The methods and techniques of optical flow discussed above, are mainly used for estimating image motion of objects in the scene. When the observed motion in the image is instead caused by camera motion and it is desired to estimate the camera motion instead of optical flow the so called direct methods are used. In direct methods of motion estimation the image velocity components in optical flow constraint (Eq. (2.15)) are substituted by the camera motion parameters and the resulting constraints are written for each pixel in the image. This results in a highly overdetermined system of equations as this constraint can be written for each pixel in the image while the number of unknown parameters are only five (instead of six as scale is not known). The derivation of such a method is given in Chapter 4 which forms the basis of the camera motion estimation algorithm presented in this thesis.

Range Flow

Range flow is the 3D motion from range image sequences. Therefore, as with optical flow, range flow is typically studied in context of high temporal sampling and it includes the notion of time. Consider a surface $Z = f(X, Y, t)$, which is a scalar function of the coordinates X and Y. Similar to the brightness constancy assumption, owing to the local rigidity of the surface, the following relationship holds:

$$Z(X, Y, t) = Z(X + \dot{X}, Y + \dot{Y}, t + 1) - \dot{Z} \tag{2.16}$$

Here, $\dot{X}, \dot{Y}$ and $\dot{Z}$ are 3D velocity components, and $t + 1$ denotes the next time instant. Using the Taylor series expansion, the following relation is obtained

$$Z(X, Y, t) + \dot{Z} = Z(X, Y, t) + \frac{\partial Z}{\partial X}\dot{X} + \frac{\partial Z}{\partial Y}\dot{Y} + \frac{\partial Z}{\partial t} + ..., \tag{2.17}$$

Here, using a linear approximation of surface i.e. approximating the surface as planar patches and neglecting the higher order terms, the range flow constraint is obtained [174].

$$\dot{Z} = \frac{\partial Z}{\partial X}\dot{X} + \frac{\partial Z}{\partial Y}\dot{Y} + \frac{\partial Z}{\partial t} \tag{2.18}$$

$$\dot{Z} = Z_X\dot{X} + Z_Y\dot{Y} + Z_t. \tag{2.19}$$

The term range flow first appeared in the work of Yamamoto et al. [208], where range flow was computed on deformable surfaces. The range flow constraint in Eq. (2.19) has also been called Elevation rate constraint equation [76] as the elevation maps of the terrain represented in the Cartesian coordinate system aligned with local terrain vertical were used to form this constraint. Harville et al. [67] have called this as depth change constraint equation.

As with the optical flow constraint, range flow constraint is based on linear approximation and therefore, it is suitable for small motions. The coarse to fine strategy should be adopted for estimating larger motions. It is important to mention here, that the surface is defined over a regular grid, image or raster, which results in 2.5D representation. For estimating spatio-temporal derivatives of the surface, the derivative filters used in optical flow constraint are also applied here.

Range flow constraint gives one constraint for each pixel and contains three unknown flow vector components. Thus, the techniques used for solving optical flow using information from the neighborhood can also be applied here. Furthermore, the aperture problem is also present in range flow estimation. In presence of planar and linear structures, only the velocity component normal to planar and linear structures can be recovered, while components parallel to these structures are ambiguous [175, 177]. Therefore, to estimate full 3D dense range flow Spies et al. [177], presented a global regularization scheme to obtain dense smooth flow vectors. In the first step a Lucas-Kanade [117] type local range flow estimation was implemented and in the second step a variational framework was used to perform a global regularization using the estimated flow vectors along with their accuracy added to a smoothness constraint.

The range flow constraint given in Eq. (2.19) contains derivatives in object space and not in image space as compared to the optical flow constraint given in Eq. (2.15). Computing the derivatives with derivative filters requires regular spacing of the samples. In case of a raster of digital terrain model, the samples of the surface are in a regular grid. However, in images from ToF camera or Kinect the depth observations or samples are unevenly sampled in the object space [174] because the object points will be farther away from each other as the depth increases. Spies and Barron [174], have presented a least squares based approach for derivative estimation of the unevenly sampled data but this approach is computationally expensive. An alternative approach is to derive the range flow constraint in image coordinates so that the well known derivative filters can be applied in image space. The surface $Z = f(x, y, t)$ is observed in the image coordinates (x, y) with a regular sampling over the image grid. Taking the derivative of $Z = f(x, y, t)$, the following relationship can be written:

$$\frac{dZ}{dt} = \frac{\partial Z}{\partial x}\frac{dx}{dt} + \frac{\partial Z}{\partial y}\frac{dy}{dt} + \frac{\partial Z}{\partial t} \tag{2.20}$$

$$\dot{Z} = Z_x\dot{x} + Z_y\dot{y} + Z_t. \tag{2.21}$$

Here in Eq. (2.21) the derivatives are in the image space and the unknowns are $\dot{x}, \dot{y}$ and $\dot{Z}$. Therefore, the derivative filters in image space can now be applied to compute Z_x and Z_y in Eq. (2.21). It should be noted that $(\dot{x}, \dot{y})$ give the velocity in the image space as compared to the velocity $(\dot{X}, \dot{Y})$ in object space as given in Eq. (2.19). In Chapter 4, Eq. (2.21) has been adapted to estimate full 3D velocity in object space as most often the interest lies in the 3D motion in the object space. Horn and Harris [76] also presented a formulation of the range flow constraint directly in sensor coordinate system and called it as range rate constraint equation.

Range flow has been mainly employed in a 2.5D framework. Where a regular grid of depth or height above ground are given. A least square solution is computed by minimizing the depth differences or height differences at each grid location. ICP and LSM on the other hand operate

on full 3D data and 3D point to point or point to plane distance is minimized. In [153] LSM is applied to 2.5D raster for laser scanning strip adjustment and the resulting formulation is the same as the range flow in Eq. (2.19).

2.4 Fusion of Range and Intensity Data

The presence of range and intensity information in laser scanning data and the low cost 3D sensors like ToF cameras and Microsoft Kinect is especially beneficial for the task of motion estimation. In case of the known depth of image points the solution of intensity image based direct methods [78] can be computed directly as it is not required to recover depth. When using only intensity images, the scale of 3D structure and motion cannot be computed. Therefore, integrating the rich texture information from intensity images with geometric information from range images, more robust methods for motion estimation can be derived.

The formulation of optical flow and range flow is principally very similar, therefore they are well suited for integration into a common framework. The optical flow constraint (Eq. (2.15)) and range flow constraint (Eq. (2.21)) share the same unknown image velocity components $\dot{x}$ and $\dot{y}$. Therefore, several authors have integrated them into a common estimation framework for motion estimation [51, 56, 149, 163, 164, 176]. Similarly the direct methods based on intensity and range images can be integrated to determine both translation and rotation components of motion. Harville et al. [67] and Jones [89–91] have combined the two constraints for estimating motion.

When the distance information is available, the image coordinates of the sparse features points can be represented in 3D in a local coordinate system and the closed form solutions discussed in Section 2.2 can be used for determining the transformation parameters. Droeschel et al. [35] and May et al. [124] have used this method to estimate the ego motion of ToF camera by matching features in consecutive frames and using the depth information of these points to compute the relative pose between the two frames. Henry et al. [72] have also used a similar technique for initialization of the ICP algorithm and then performed a joint optimization of sparse feature correspondences and ICP to estimate the transformation between consecutive frames of Kinect. The use of range sensors in SLAM methods is also very advantageous as it can lead to faster and accurate determination of camera pose along with the dense environment map. The RGBD SLAM algorithm of Endres et al. [37] finds visual feature points in color images and then using the corresponding depth of these feature points a pairwise transformation is between two frames is computed. This procedure is performed for a subset of images.

The method of relative orientation presented in this thesis is similar to the work of Harville [67] but it is extended to estimate large motions and include robust estimation. The method of estimating motion of independently moving object is closely related to that of Spies et al. [177], which is extended to include anisotropic smoothing in a linear least squares regularization scheme.

CHAPTER 3

Range Measuring Sensors

3.1 Time of Flight Cameras

Time-of-Flight (ToF) cameras or sometimes also known as *Range IMaging cameras* (RIM) are based on active optical measuring technology to capture 3D information at each pixel [150]. These cameras indirectly measure the time of flight by computing the phase difference between an emitted and the received signal. The phase difference is computed by the cross correlation of the received and the emitted signal. Most ToF cameras illuminate the scene with a continuous wave sinusoidal or a rectangular signal, modulated over an infrared light source [23, 166]. The demodulation or cross correlation of the received signal with the emitted signal is performed simultaneously at each pixel using the so called demodulating or lock in pixels [23, 112]. The demodulation process for a sinusoidal signal consists of sampling the received signal at 0^o, 90^o,180^o and 270^o phases and based on these samples the phase difference ϕ_d, amplitude A and offset B as shown in Figure 3.1 are then computed by [23]:

$$\phi_d = atan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \tag{3.1}$$

$$A = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \tag{3.2}$$

$$B = \frac{A_0 + A_1 + A_2 + A_3}{4} \tag{3.3}$$

where A_0,A_1,A_2 and A_3 are the measurements of the returned signal at 0^o, 90^o,180^o and 270^o phase differences respectively. The sampling of signal at these phases can be achieved by either 1-tap, 2-tap or 4-tap pixel architectures. A 1-tap pixel will sample the signal only at one phase at one time, therefore, 4 separate exposures should be performed to compute one distance measurement, which leads to low frame rate and can cause motion blur for fast moving objects. In contrast 2-tap and 4-tap pixels require two exposures and a single exposure respectively [98].

Capturing more samples at the same instant (as in case of 4-tap) pixels comes at cost of low fill factor and different channel characteristics within the pixel for each sample [23]. From the measured phase difference ϕ_d, range is computed as [23]

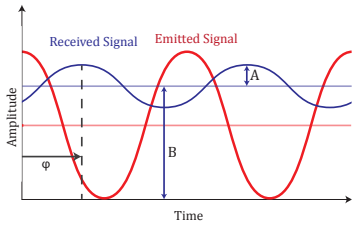

Figure 3.1: Sinusoidal emitted and received signal in ToF cameras

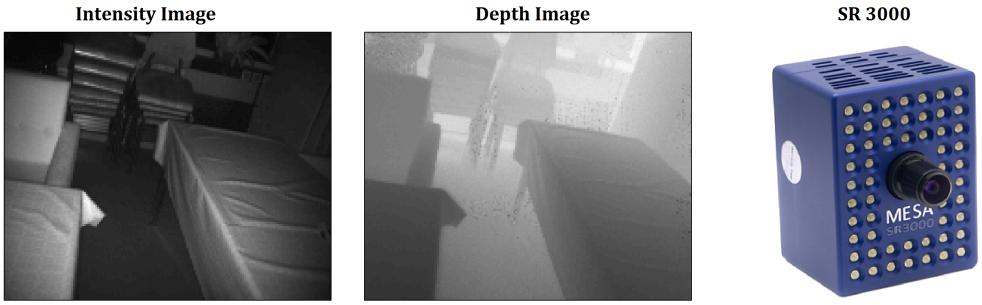

Figure 3.2: *Left* and *Middle*: Intensity and range images from the camera. *Right*:SR3000 ToF camera used for experiments in this thesis.

$$D_R = \frac{c \cdot \phi_d}{4 \cdot \pi \cdot f_{mod}} \tag{3.4}$$

Here c is the speed of light and f_{mod} is the modulation frequency. For a modulation frequency of 20 MHz, the maximum unambiguous range is 7.5 meters. Distances larger than the maximum unambiguous range are phase wrapped. Typical unambiguous range of the ToF cameras is from 5 to 10 meters. A phase unwrapping procedure [92] can be applied to recover distances larger than the maximum unambiguous range.

The stochastic properties of the measured range (Eq. (3.4)) mainly depend on amplitude of the returned signal, photon shot noise and dark current. The standard deviation of the range measurement can be derived using error propagation rule on Eq. (3.4) [23, 111]:

$$\sigma_d = \frac{c}{4\sqrt{2}\pi f_{mod}} . \frac{\sqrt{B+A}}{A} \tag{3.5}$$

Here, B is the offset, which also contains the background light as shown in Figure 3.1. A more detailed statistical analysis of measurements in ToF cameras is given in [133]. The accuracy

(σ_d) of the estimated range is inversely proportional to the modulation frequency f_{mod}. Thus, better range accuracy can be achieved using higher modulation frequencies. However, the unambiguous range decreases with increasing f_{mod}. Therefore, better accuracy comes at the cost of shorter unambiguous range. Amplitude of the received signal is another important factor in determining the accuracy of the measured distance. A typical scene consists of surfaces at different distances from the camera and having varying light reflecting properties. The strength of the emitted signal decreases with inverse square law. Therefore, objects which are farther away from camera and have low light reflecting characteristics, will show a low distance accuracy. This problem can be partially avoided by increasing the strength of the emitted signal which will lead to increase in signal to noise ratio. However, the emitted signal should comply to eye safety regulations, therefore the strength of the signal cannot be increased beyond certain limit. Another solution for better accuracy is to increase the integration time i.e. averaging the received signal over longer time to reduce the noise. However, a higher integration time results in lower frame rate and may also result in motion blur and pixel saturation. Therefore, the amount of integration time which gives a good comprise between signal to noise ratio, motion blur and saturation is often empirically selected based on the experimental requirements. As given in Eq. (3.5), the background signal level also decreases the precision of the depth measurement. Sunlight can contribute to a large portion of this background signal. The sunlight not only increases the background signal but can also lead to pixel saturation due to limited dynamic range of the pixels. Therefore, direct sunlight should be avoided, which is one reason why ToF cameras are mainly suited for indoor application. One focus of research in ToF cameras design is higher background light subtraction [30] and higher dynamic range [33].

The systematic errors in addition to random errors originate from factors like object reflectivity, integration time and object distances. Therefore, a calibration for correction of systematic errors in the distance measurement is usually applied. More detail of these systematic errors and the calibration procedures for the compensation of these errors can be found in [94, 96, 113]. Further distortions in the distance measurement may arise from scattering [97, 114] and multipath effect [45]. Scattering is caused by internal reflections inside camera body and lens. For a given camera the magnitude of the distortion highly depends on the depth variations in the scene. Multipath effect is caused by reflections from corners or intersecting surfaces. Keeping all these error sources in mind, the experimental setup is designed in a way to reduce the effect of these errors.

In this thesis SR3000 [127] ToF camera is used, which is based on 2-tap pixel architecture [23]. It has a resolution of 144×176 pixels, a maximum frame rate of 25 fps and a field of view of 39.6×47.5 degrees . A view of an SR3000 camera along with a sample intensity image and range image is shown in Figure 3.2.

3.2 Active Triangulation Cameras

Active triangulation cameras or also known as structured light cameras are another type of cameras capable of acquiring 3D information. These cameras measure the distance by triangulating a pattern projected on the scene. In passive triangulation, depth is estimated by matching image texture between the two images of a stereo pair. The estimation of the depth depends on the

amount of texture in the images, therefore, when there is no image texture the corresponding points in the stereo pair and thus the depth cannot be determined. Active triangulation cameras avoid this problem by projecting a known pattern on the scene and then match this pattern in the image to estimate depth at each pixel. In the recent years, low cost, high frame rate active triangulation cameras like Microsoft Kinect and Asus Xtion [55, 99] have become commercially available.

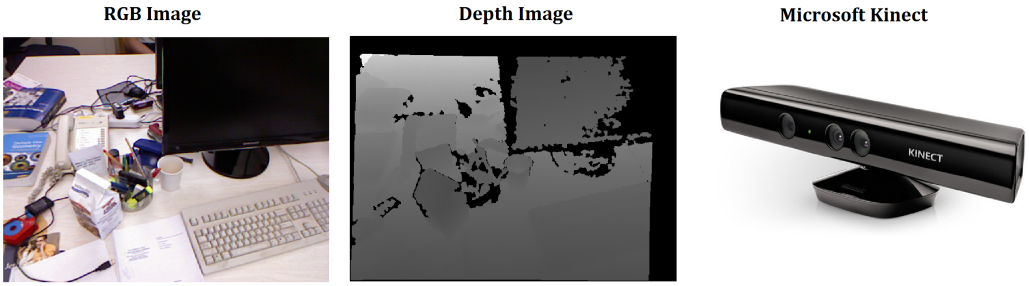

Figure 3.3: Kinect sensor, RGB and depth image from RGB-D SLAM dataset [180]

The depth estimation in Microsoft Kinect is based on structured light triangulation. Kinect consists of an RGB camera, an IR camera and an IR projector. The IR projector projects a speckle pattern on the scene, which is then matched in the IR image to generate the depth map. In [106, 171] it is stated that a correlation window of 9×9 or 9×7 is used and after further refinement, a sub-pixel accuracy of 1/8th is achieved. The disparity image of Kinect has a resolution of 640×480. Calibration of Kinect's RGB and IR cameras is given in [26, 99, 106], while a detailed photogrammetric calibration of Kinect cameras is presented in [26]. According to the accuracy model of [99], the depth accuracy decreases quadratically from a couple of millimeters at 0.5 m distance to about 4 cm distance at 5 m distance from the camera.

Figure 3.3 shows the Kinect sensor and sample RGB and depth image from Kinect. These images are from RGB-D SLAM dataset [180]. As compared to ToF cameras, the color or brightness information in Kinect is captured from a different camera. Therefore, the depth and color images need to be registered to each other. In Figure 3.3 part of the depth image is cut out because of different camera view points, so part of the RGB and depth images don't overlap. There is also missing data in the depth image due to occlusions in the stereo geometry and areas where the matching of the projector pattern wasn't successful. Furthermore, the depth of each pixel is estimated from a matching procedure using a neighborhood window, therefore, the individual depth observations are not independent of each other.

Asus Xtion also works on the same principal as Kinect and also contains an IR camera, an RGB camera and an IR projector. The depth image has a resolution of 640×480 at 30 fps. Figure 3.4 shows the Asus Xtion and a sample RGB and depth image from the RGBD SLAM dataset [180].

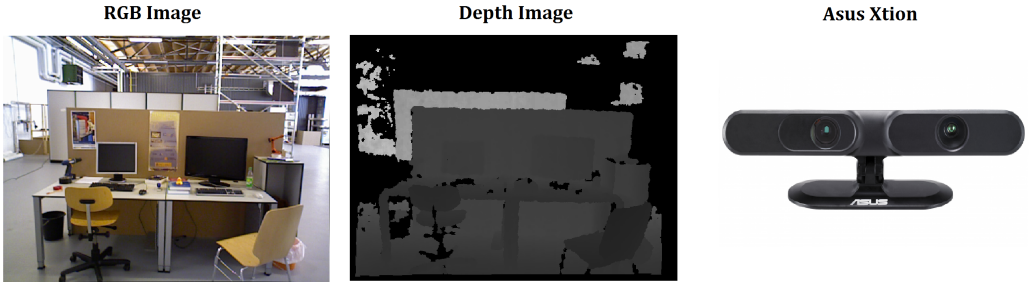

Figure 3.4: Asus Xtion sensor, RGB and depth image from RGB-D SLAM dataset [180]

3.3 Laser Scanning

Laser scanning sometimes also termed as LiDAR (light detection and ranging) is a technology to measure the distance to the object by measuring the time between an emitted pulse of light and received echo [108]. A laser beam is deflected at different angles using e.g. a rotating prism and time of flight for each laser pulse is recorded. Modern day laser scanner are able to record upto 500,000 echoes per second. Airborne laser scanning and terrestrial laser scanning are two main platforms for laser scanning systems. In airborne laser scanning (ALS) the laser beam is deflected at right angles to the flight direction which gives data in across flight direction and the movement of the airplane itself gives the data in along flight direction. Airborne laser scanning (ALS) is a popular method for topographic modeling and developing terrain models of vast areas [46, 109]. Terrestrial laser scanning (TLS) is used in applications requiring detail acquisition of selective sites and objects. In contrast to ALS the laser beam in TLS is deflected along two directions to acquire data along both horizontal and vertical directions. In TLS scans are acquired from different view points which are then co-registered using e.g. the ICP algorithm [156].

Full-waveform scanners can provide complete digitized recorded waveforms instead of discrete echoes from the returned signals. The full-waveform [197] recorded echo can then be decomposed to estimate the properties of the individual scatteres in the laser footprint [154]. The amplitude of the recorded pulse gives information about the radiometric properties of the surface and the reflectivity can be computed using the inclination angle and area of the surface [155, 197]. The laser pulse covers a very narrow band in frequency spectrum, therefore, the amplitude is typically less informative than e.g. a color image. Modern day laser scanners can work upto ranges of few kilometers and an accuracy of $1:10,000$ or better is usually achievable for the maximum recommended range of the laser scanner.

In comparison to ToF cameras and Kinect, which simultaneously acquire distance information over the entire image plane, laser scanner record distance data sequentially and gives a less structured point cloud instead of data over a regular grid. The spatial resolution of the data can be estimated using point density measure. The ALS data e.g. is characterized by providing the number of points over a grid cell e.g. 4 $points/m^2$. The principle of measurement is however, similar to ToF camera in the sense that the distance is acquired using time of flight of the emitted signal and the intensity is related to the amplitude of the received signal. Figure 3.5, shows a

sample TLS scan of the landslide discussed in Chapter 6.

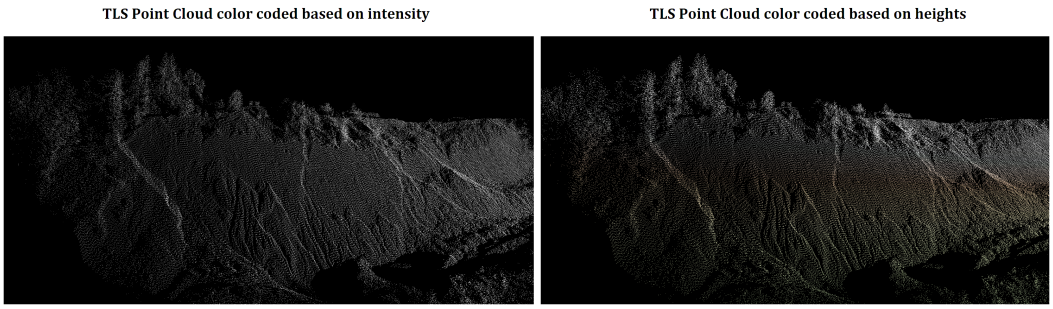

Figure 3.5: TLS scan of the scarp of a landslide area, *Left*: Gray levels according to the intensity *Right*: Color coded according to the height

CHAPTER 4

Relative Orientation and Bundle Adjustment

In this chapter, a method for determining the orientation of a moving camera is presented. First the relative orientation of consecutive image pairs is estimated using range flow and optical flow constraints and then these relative orientation results are used in bundle adjustment to determine the orientation of a large number of images in a common coordinate system. Following up with the theory of image orientation as given in Chapter 2, Figure 4.1 shows a point $\mathbf{P}$ observed from three camera positions. The coordinate systems (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) and (X_3, Y_3, Z_3) are aligned with the corresponding image coordinate systems (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3) and placed at the projection centers of these three camera positions. The transformation between the coordinates of point $\mathbf{P}$ as observed in the coordinate systems (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) is given as:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}; \quad \mathbf{X}_2 = \mathbf{R}_{21}\mathbf{X}_1 + \mathbf{T}_{21} \tag{4.1}$$

The translation $\mathbf{T}_{21}$ and rotation $\mathbf{R}_{21}$ gives the orientation parameters of the first image relative to the second image. The translation $\mathbf{T}_{21}$ is a vector from origin of (X_2, Y_2, Z_2) to origin of (X_1, Y_1, Z_1), while the rotation matrix $\mathbf{R}_{21}$ aligns the coordinate system (X_1, Y_1, Z_1) with (X_2, Y_2, Z_2). Similarly, the relative transformation between the second and third image can be written as:

$$\mathbf{X_3} = \mathbf{R}_{32}\mathbf{X_2} + \mathbf{T}_{32} \tag{4.2}$$

where, $\mathbf{T}_{32}$ and $\mathbf{R}_{32}$ are the orientation parameters of the second image relative to the third image. If more images are available the relative orientations can be written similarly for each pair. Section 4.1 presents a method to estimate relative orientation parameters as given in Eqs. (4.1) and (4.2) using range flow and optical flow constraints. The relative orientation as given in Eqs. (4.1) and (4.2) are in different coordinate systems. However, to estimate the motion or

trajectory of the camera in a video it is desired to compute the position and orientation of each camera with respect to one common coordinate system. Transforming all relative orientations to a common coordinate system does not give a globally optimal solution as it will result in accumulation of errors. Therefore, to achieve a globally optimal solution of camera motion, bundle adjustment is performed using sparse feature matching and pairwise relative orientation constraints, which is presented in Section 4.2.

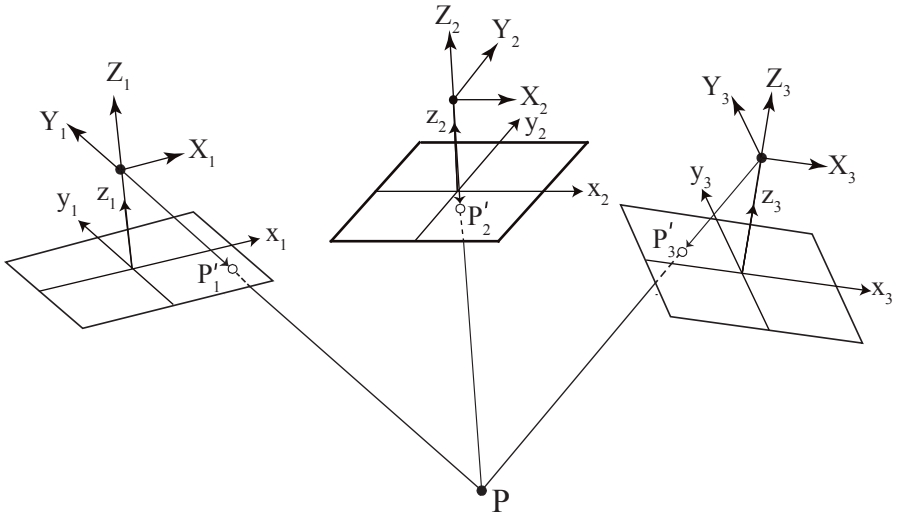

Figure 4.1: A point $\mathbf{P}$ observed in three images. The coordinate systems (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) and (X_3, Y_3, Z_3) are placed at the projection centers of the three images and aligned with the corresponding image coordinate systems (adapted from [108])

As relative orientation is specified between two image-aligned coordinate systems whereas bundle adjustment relates image coordinates to a superior coordinate system, further consideration of the parameterizations of 3D rotations is required. Any rotation in 3D can be described by three angles also known as Euler angles, therefore, there are three unknowns corresponding to a rotation in 3D. Each rotation by an angle can be written in form of a matrix and three rotation matrices can be written correspondingly to each angle and these matrices can then be multiplied to obtain a single rotation matrix that can perform any rotation in 3D space. The rotation matrix is a 3×3 matrix containing three unknowns and nine elements. This means that the nine elements of the rotation matrix are not independent and these elements should satisfy several constraints to form a rotation matrix. The three columns of a rotation matrix are unit vectors and are orthogonal to each other. Therefore, there are three normalization constraints and three orthogonality constraints that the elements of a rotation matrix need to satisfy. Due to the fact that the three columns of a rotation matrix are orthogonal unit vectors its determinant is one. The rotation matrix $\mathbf{R}$ in Eq. (2.1) rotates the coordinate system (X, Y, Z) to align with coordinate system (X', Y', Z'). The selection of axes, around which the three rotations are executed can also vary, this leads to different parameterization of the rotation matrices. A commonly used

parameterization of rotation matrix in aerial photogrammetry based on rotation angles *omega*, *phi* and *kappa* (ω, φ, κ) is given as [108].

$$\mathbf{R}_{\omega\varphi\kappa} = \begin{pmatrix} c\varphi c\kappa & -c\varphi s\kappa & s\varphi \\ c\omega s\kappa + s\omega s\varphi c\kappa & c\omega c\kappa - s\omega s\varphi s\kappa & -s\omega c\varphi \\ s\omega s\kappa - c\omega s\varphi c\kappa & s\omega c\kappa + c\omega s\varphi s\kappa & c\omega c\varphi \end{pmatrix} \tag{4.3}$$

Here, ω is the primary rotation around X axis, φ is the secondary rotation around rotated Y and κ is the tertiary rotation around rotated Z axis as in Figure 2.1, c and s are the cosine and sine of angle respectively. Given a rotation matrix, individual angles $(\omega, \varphi, \kappa)$ of rotation matrix can be computed from elements of rotation matrix [108]. *Alpha*, *zeta* and *kappa* is a commonly used rotation matrix parameterization [107] in terrestrial photogrammetry. The parameterization of 3D rotations using rotation matrices can result in singularities the so called gimbal lock problem. To avoid these singularities different parameterization of 3D rotations can be used like axis-angle and quaternion representations. The fact that one of the eigenvalue of $\mathbf{R}$ is one, means that the rotation can be represented by a vector and a single rotation around this vector, which leads to axis-angle representation of the rotation. Quaternions represent rotations using four parameters and there exist one constraint among the parameters. A more detailed description of parameterizing rotations in 3D space can be found in following texts [126, 168].

If the linearization of the collinearity equations is performed by directly using the rotation matrix in Eq. (4.3), then the normalization and orthogonality constraints do not need to be specified. Furthermore, as in this work the choice of the reference coordinate system is arbitrary, the reference system can be chosen to avoid singularity. In this work the $(\omega, \varphi, \kappa)$ parameterization (Eq. (4.3)) is used for representing rotations although in principal any other rotation parameterization could be used.

Now, if the motion between two camera positions is small (as it is typically the case in videos), the three rotations angles in $\mathbf{R}_{\omega\varphi\kappa}$ are also small. When these angles are small, the approximations $\lim_{\omega\to 0} cos(\omega) = 1$ and $\lim_{\omega\to 0} sin(\omega) = \omega$ can be used to approximate the rotation matrix (Eq. (4.3)) as:

$$\mathbf{R}^S = \begin{bmatrix} 1 & -\kappa & \varphi \\ \kappa & 1 & -\omega \\ -\varphi & \omega & 1 \end{bmatrix} = \begin{bmatrix} 1 & -R_Z & R_Y \\ R_Z & 1 & -R_X \\ -R_Y & R_X & 1 \end{bmatrix} \tag{4.4}$$

Eq. (4.4) gives the small angle approximation of the rotation matrix. The superscript *s* in $\mathbf{R}^S$ has been used to show that its a small angle approximation. The angles R_X, R_Y, R_Z represent the rotations around X, Y and Z axes respectively (as in Figure 2.1) which means that the three angles can be written directly according to the axis on which the rotation was performed, this also means that the small angle approximation of rotation matrix is independent of the original parameterization if the three rotations were performed on three independent axis.

4.1 Relative Orientation using Optical Flow and Range Flow

The method of determining relative orientation used here utilizes optical flow and range flow constraints. These type of methods are known as *direct methods* as they determine the unknown

parameters based on directly measured image quantities such as intensity and depth [78, 85] without computing *flow* or finding feature points. The other commonly used method of determining the relative orientation are based on feature correspondences in the images. Due to the fact that optical flow constraint and range flow constraint are principally similar, they can be well integrated in an estimation problem. Utilizing dense range and image information is especially advantageous for estimating motion in scenes with low geometric structure and radiometric texture.

Now the method of estimating relative orientation of a moving camera in range and intensity image sequence is derived. Figure 4.1 shows the point $\mathbf{P}$ measured from three camera positions. The motion between consecutive images of the sequence is assumed to be small but for sake of clarity Figure 4.1 shows the image positions with relatively large motion. The $3D$ coordinates of a point $\mathbf{P}$ as measured in the first image (X_1, Y_1, Z_1) and the second image (X_2, Y_2, Z_2) are related by:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} 1 & -R_Z & R_Y \\ R_Z & 1 & -R_X \\ -R_Y & R_X & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \tag{4.5}$$

here, the small angle approximation of the rotation matrix is used, as it is assumed that the motion between the camera positions is small. $(T_X, T_Y, T_Z, R_X, R_Y, R_Z)$ are the unknown parameters of the relative orientation between the two camera positions. Now, the change in the $3D$ coordinates are given as:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{bmatrix} = \begin{bmatrix} 0 & -R_Z & R_Y \\ R_Z & 0 & -R_X \\ -R_Y & R_X & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \tag{4.6}$$

The Eq. (4.6) is the differential form of Eq. (4.5) as it gives the change in the $3D$ coordinates of the measured point. Let (x_1, y_1) be the image coordinates of object point (X_1, Y_1, Z_1). As the coordinate system (X_1, Y_1, Z_1) is chosen to be aligned with (x_1, y_1, z_1), using the perspective projection model, following relationship exists between the object and the image coordinates:

$$x_1 = f\frac{X_1}{Z_1} \qquad y_1 = f\frac{Y_1}{Z_1} \tag{4.7}$$

Similarly, in the inverse form, the $3D$ object coordinates can be written in terms of image coordinates as

$$X_1 = x_1\frac{Z_1}{f} \qquad Y_1 = y_1\frac{Z_1}{f} \tag{4.8}$$

For sake of simplicity the index of the coordinates is removed. Using the inverse mappings of (X, Y), the change in the $3D$ coordinates in Eq. (4.6) can be written as:

$$\dot{X} = -R_Z Y + R_Y Z + T_X = -yR_Z\frac{Z}{f} + R_Y Z + T_X \tag{4.9}$$

$$\dot{Y} = R_Z X - R_X Z + T_Y = xR_Z\frac{Z}{f} - R_X Z + T_Y \tag{4.10}$$

$$\dot{Z} = -R_Y X + R_X Y + T_Z = -xR_Y \frac{Z}{f} + yR_X \frac{Z}{f} + T_Z \tag{4.11}$$

The relation between the 2D image velocity and the 3D object velocity is derived by first order approximation of Eq. (4.7):

$$\dot{x} = \frac{f}{Z}\dot{X} - \frac{fX}{Z^2}\dot{Z} \qquad \dot{y} = \frac{f}{Z}\dot{Y} - \frac{fY}{Z^2}\dot{Z} \tag{4.12}$$

$$\dot{x} = \frac{f}{Z}\dot{X} - \frac{x}{Z}\dot{Z} \qquad \dot{y} = \frac{f}{Z}\dot{Y} - \frac{y}{Z}\dot{Z} \tag{4.13}$$

It should be noted that Eq. (4.12) is based on first order approximation, therefore, it is valid for small motions. Finally, by also considering the expressions for the 3D velocities (Eqs. 4.9–4.11) the 2D velocities become:

$$\dot{x} = \frac{fT_X}{Z} - \frac{xT_Z}{Z} - R_X \frac{xy}{f} + R_Y \left(f + \frac{x^2}{f}\right) - yR_Z \tag{4.14}$$

$$\dot{y} = \frac{fT_Y}{Z} - \frac{yT_Z}{Z} - R_X \left(f + \frac{y^2}{f}\right) + R_Y \frac{xy}{f} + xR_Z \tag{4.15}$$

Eqs. 4.14 and 4.15 give the relationship between image velocity and camera translation and rotation assuming that the observed motion is due to motion of the camera. Now that the relationship between unknown motion parameters and the image velocity is obtained, optical flow and range flow constraints can be utilized which can determine the image velocity using intensity and depth information. The optical flow constraint and range flow constraints are written as:

$$I_x \dot{x} + I_y \dot{y} = -I_t \tag{4.16}$$

$$\dot{Z} = Z_x \dot{x} + Z_y \dot{y} + Z_t \tag{4.17}$$

Inserting $\dot{Z}$ from Eq. (4.11) in to the range flow constraint gives:

$$Z_x \dot{x} + Z_y \dot{y} - T_Z - y\frac{Z}{f}R_X + x\frac{Z}{f}R_Y = -Z_t \tag{4.18}$$

The image velocities $(\dot{x}, \dot{y})$ from Eqs. (4.14) and (4.15) can be inserted into the optical flow (Eq. (4.16)) and range flow (Eq. (4.18)) equations to give the relation between the 6D camera transformation parameters and the observed range and intensity images.

$$\begin{bmatrix} I_x\frac{f}{Z} & I_y\frac{f}{Z} & -\left(I_x\frac{x}{Z} + I_y\frac{y}{Z}\right) \\ Z_x\frac{f}{Z} & Z_y\frac{f}{Z} & -\left(Z_x\frac{x}{Z} + Z_y\frac{y}{Z} + 1\right) \end{bmatrix} \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} +$$
$$\begin{bmatrix} -\left(I_x\frac{xy}{f} + I_y\frac{f^2+y^2}{f}\right) & \left(I_y\frac{xy}{f} + I_x\frac{f^2+x^2}{f}\right) & (I_y x - I_x y) \\ -\left(Z_x\frac{xy}{f} + Z_y\frac{f^2+y^2}{f} + y\frac{Z}{f}\right) & \left(Z_y\frac{xy}{f} + Z_x\frac{f^2+x^2}{f} + x\frac{Z}{f}\right) & (Z_y x - Z_x y) \end{bmatrix} \begin{bmatrix} R_X \\ R_Y \\ R_Z \end{bmatrix} = \begin{bmatrix} -I_t \\ -Z_t \end{bmatrix} \tag{4.19}$$

Eq. (4.19) gives a linear relationship between camera orientation parameters and the change in image intensity and depth at each pixel. The only quantities required in this relationship are the spatio temporal derivatives of intensity and depth. These derivatives can be computed efficiently using derivatives filters. In Eq. (4.19) most of the coefficients contain intensity and depth derivatives, however, (T_Z, R_X, R_Y) contains terms which are independent of derivatives, which means that if e.g. the scene consists of only a plane parallel to the image plane with homogeneous gray level, motion components (T_Z, R_X, R_Y) can still be determined even though (I_x, I_y, Z_x, Z_y) are all zero.

Estimating image derivatives is an essential part of *flow* estimation, as the spatial and temporal derivatives appear in the optical flow and range flow constraints. In digital image only discrete samples of the function i.e. intensity and range are observed, therefore, the derivatives are approximated for a discrete case. In Horn and Schunck optical flow [77], eight point derivatives based on a cube of two images is used for estimating spatial and temporal derivatives of image intensity. Simoncelli [169] proposed matched filters for computing spatio-temporal derivatives consisting of low pass pre-filters and derivative filters. The derivatives filters are designed in such a way that they are good approximation of the low pass pre-filter. The 5-point derivative filter $\frac{1}{12}\begin{bmatrix}-1 & 8 & 0 & -8 & 1\end{bmatrix}$ is often used for computing derivatives [10, 182, 204]. In this work, the 5-point Simoncelli spatio temporal filters are used when the motion is small and continuous i.e. no sudden acceleration. Using these central differences derivatives means that the derivatives are computed at the subject pixel, whereas in forward or backward difference filter kernels the derivative is estimated in the middle of pixels like the eight point filter given by Horn and Schunck [77]. In the 5-point Simoncelli filters, five images are used for computation of both spatial and temporal derivatives, which assumes that the motion in these five images is similar. However, if there are abrupt changes in motion or if the data is not continuous in time (e.g. the landslide case, presented in Chapter 6), then only two images are used for computing spatial derivatives using the filter $\frac{1}{12}\begin{bmatrix}-1 & 8 & 0 & -8 & 1\end{bmatrix}$ filter and temporal derivative is computed as the difference of corresponding pixels, this means that the temporal derivative is described as the change between the value of intensity or depth at the corresponding pixel.

To determine the relative orientation of an image pair, two observation equations using Eq. (4.19) can be written for each image pixel. This leads to a highly over determined equation system, which can be solved using least squares. To perform least squares adjustment, Eq. (4.19) for all pixels can be written in the form:

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{l} + \mathbf{e} \tag{4.20}$$

where $\mathbf{A}$ contain the coefficients of depth constraint and intensity constraint as in Eq. (4.19) respectively. The observations $\mathbf{l}$ are the time derivatives of depth and intensity respectively as in Eq. (4.19) and $\mathbf{e}$ contains the residuals for each observation equation. If $\mathbf{Q}$ is the covariance matrix of the observation, then $\mathbf{P}_l = \mathbf{Q}^{-1}$ is the weight matrix and the least squares solution is given by:

$$\boldsymbol{\beta} = (\mathbf{A}^T\mathbf{P}_l\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{P}_l\mathbf{l}) \tag{4.21}$$

The weight matrix $\mathbf{P}_l$ determines the weighting of each observation equation in the adjustment. Here, it is assumed that the observations are independent of each other which implies that

the matrix $\mathbf{Q}^{-1}$ and $\mathbf{P}_l$ are diagonal matrices. In ordinary least square solution as in Eq. (4.21) it is assumed that only one depth and intensity measurement in each Eq. (4.19) is corrupted by noise. Therefore, each of these observations can be weighted according to its *a priori* variance. The intensity values and range values in the image are in different units, therefore, weighting them by *a priori* variances will result in the homogenization of the equations [107]. Some authors have used a relative weighting factor to weight the whole group of intensity and the depth observations by a single parameter [89, 177]. The variance of the estimated parameters is given by:

$$\mathbf{Q}_\beta = \sigma_0^2(\mathbf{A^T Q^{-1} A})^{-1}, \tag{4.22}$$

$$\sigma_0 = \sqrt{\frac{\mathbf{e}^T\mathbf{P}_l\mathbf{e}}{n-6}}, \quad \mathbf{e} = A\boldsymbol{\beta} - \mathbf{l}, \tag{4.23}$$

Here, σ_0 is the standard deviation of the adjustment and $\mathbf{Q}_\beta$ gives the variances of each estimated parameters. n is the number of observations and $n-6$ is the redundancy as there are six unknown parameters.

Eq. (4.19) is true for pixels that do not contain any independently moving object. Therefore, in presence of some independently moving object robust adjustment is necessary to remove the effect of outliers i.e. pixel belonging to an independently moving object. This is one advantage of using dense image information, that in presence of the outliers or independently moving object, robust estimation will converge to the dominant motion or global motion, which in this work is assumed to be generated due to camera motion [85]. Robust adjustment is performed by iteratively re-weighted least squares. A robust weighting function like *talwar* [74] as given in Eq. (4.24) is used for weighting of each observation based on the corresponding residual from previous iteration. In the case of Eq. (4.24) the observation equation whose normalized residual e_n is greater than some threshold λ, gets a weight of zero for the next iteration. The threshold λ can be chosen, so that the normalized residual greater than five standard deviations is regarded as an outlier and assigned a weight of zero.

$$w = \begin{cases} 0 & \text{if } abs(e_n) > \lambda \\ 1 & \text{if } abs(e_n) \leq \lambda \end{cases} \tag{4.24}$$

Eq. (4.19) is derived using first order Taylor approximation, therefore, it is suitable for small motions. For larger motions, coarse to fine strategy is applied. Orientation parameters are first estimated at a coarser resolution. Intensity image is warped i.e. interpolated using the motion given by Eqs. 4.14 and 4.15. The depth image is first translated according to Eq. (4.11) and then interpolated using the motion given by Eqs. 4.14 and 4.15. This step is repeated till the parameters are computed at the finest resolution. If $[L1, L2, L3]$ are the three image resolutions with $L1$ being the coarsest resolution and $L3$ being the finest resolution, the final transformation or the relative orientation using all the image resolutions is:

$$\begin{aligned} \mathbf{T} &= \mathbf{T}_{L1} + \mathbf{R}_{L1}\mathbf{T}_{L2} + \mathbf{R}_{L1}\mathbf{R}_{L2}\mathbf{T}_{L3} \\ \mathbf{R} &= \mathbf{R}_{L1}\mathbf{R}_{L2}\mathbf{R}_{L3}, \end{aligned} \tag{4.25}$$

where $[T_{L1}, T_{L2}, T_{L3}]$ and $[R_{L1}R_{L2}R_{L3}]$ are translational and rotational parameters at $[L1, L2, L3]$ resolutions respectively.

4.2 Bundle Adjustment with Relative Orientation Constraints

In section 4.1, a method for computing relative orientation of an image pair was presented. If a sequence of images is available and the task is to compute the camera motion during the whole sequence, then a simple solution is to compute relative orientation of each consecutive image pair and transform all these orientations into a common coordinate system. The problem in such a solution is that errors will accumulate from first till the last image. Therefore, it is necessary to utilize more scene information to achieve a globally consistent solution. In this section a method using relative orientation constraints in the bundle adjustment is presented, which gives a globally consistent solution to the camera orientation problem.

Bundle adjustment is used to compute the orientation of block of images along with 3D position of sparse feature points. Each feature point in one image gives two observation equations, which are the collinearity equations (Eq. (2.3)) as presented in Chapter 2. SIFT or SURF features are used to find corresponding points in the images. Feature matching is an essential step in obtaining globally optimal solutions and reducing the drift and accumulation of the errors, as identification of loop closures or revisit (re-capture) of a scene area and matching of corresponding points over longer baselines is achieved using robust feature matching [39, 40, 141]. The feature matching is performed using the *keyframes* strategy as is often employed in SLAM algorithms [38, 72]. The *keyframes* are selected as follows: the first image in the sequence is always a *keyframe* and each new image is matched to this *keyframe*. If an image cannot be matched to this *keyframe* it is defined as the new *keyframe*. So each new image is only matched to one of the *keyframes*. This strategy is important in order to avoid matching each frame to every other frame, as it will be computationally very expensive.

The orientation of each image and the coordinates of each point in bundle adjustment are typically given in a common reference frame or coordinate system. Figure 4.2 shows a point measured in three images with each image having its own coordinate system. The coordinate system (X, Y, Z) can be regarded as a common or global coordinate system and it is desired to compute the orientation of each image and object point relative to this reference coordinate system. Determination of this reference system is known as datum definition. Seven constraints i.e. three translations, three rotations and one scale factor are required to fully define this datum, because without defining this datum, the whole block can be translated, rotated and scaled while satisfying the observation equations [14, 44, 146]. The datum is commonly defined using control points. In this thesis, however, no global information in terms of control points or global positioning system is assumed, so it is convenient to attach the reference coordinate system with the orientation of the first camera. Therefore, the projection center of the first camera is at location $(0, 0, 0)$ and the first image is aligned with the axes of the global coordinates system and the projection center of each camera frame and the 3D coordinates of features points are computed relative to this reference frame. The scale is determined by the range measurements in the range cameras, which can be written as an additional observation equation [113] for each

point. If $(X_{O_1}, Y_{O_1}, Z_{O_1})$ are the coordinates of O_1 (Figure 4.2) then the range observation can be written as:

$$D_R = \sqrt{(X - X_{O_1})^2 + (Y - Y_{O_1})^2 + (Z - Z_{O_1})^2} \tag{4.26}$$

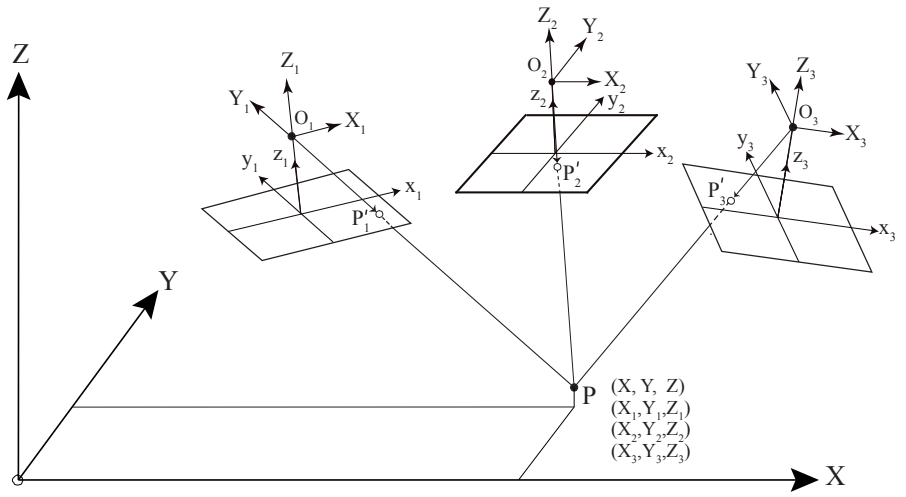

Figure 4.2: A point **P** observed in the three images and the global coordinate system (X, Y, Z) (adapted from [108]).

Hence by fixing the origin at the first camera orientation and using depth measurements, the datum is fully defined. Now, for each feature point in each image, three equations (Eqs. (2.3) and (4.26)) can be written. These equations are weighted according to their *a prior* variances in the adjustment. Both collinearity equations and depth observation equation are nonlinear, they are first linearized using an approximate solution. The differential coefficients for the linearized collinearity equations can be found in [108]. The approximate solutions for initializing bundle adjustment are computed by first matching features in images and then using the 3D location of the points to compute the transformation between the points using the closed form method presented in Chapter 2. The relative orientations computed using optical and range flow can also be transformed into the common coordinate system to serve as the approximate solution for each camera position and then approximating the 3D position of each point in this coordinate system.

In the above setup, relative orientations estimated using range and optical flow have not been used in bundle adjustment. If there is good distribution of feature points and the feature matching is robust enough, bundle adjustment with only feature points and depth measurements can accurately estimate unknown orientations (assuming that the systematic errors have been taken into account, and distance measurements are correctly modeled). However, if the features points are sparse and not well distributed in the image, the absolute accuracy can be lower. Therefore, it becomes essential to integrate further information in the adjustment. The relative orientation method based on optical flow and range flow implicitly takes into account the matching of

lines and corners. Therefore, the results of relative orientation can be used to constrain camera orientations to obtain better estimates.

Now the equations for constraining the relative orientations in bundle adjustment are given. The relative orientation computed in Eqs. (4.1) and (4.5) is defined to transform the first coordinate system to the coordinate system of the second camera position. In bundle adjustment the orientation of each frame is given in a common coordinate system. Suppose, two camera positions, the exterior orientation of these cameras positions as given in (X, Y, Z), which is now called the global coordinate system is given as:

$$\mathbf{X} = \mathbf{R}_1\mathbf{X}_1 + \mathbf{T}_1 \qquad \mathbf{X} = \mathbf{R}_2\mathbf{X}_2 + \mathbf{T}_2 \tag{4.27}$$

where $(\mathbf{R}_1, \mathbf{T}_1)$, are the exterior orientation parameters of the first image and $(\mathbf{R}_2, \mathbf{T}_2)$, are the exterior orientation parameters of the second image respectively. The relative orientation between the two positions using the transformations given in Eq. (4.27) is:

$$\mathbf{X}_2 = \mathbf{R}_2^T\mathbf{R}_1\mathbf{X}_1 + \mathbf{R}_2^T(\mathbf{T}_1 - \mathbf{T}_2) \tag{4.28}$$

The relative orientation between the first and the second camera position as given in Eq. (4.5) is of the form:

$$\mathbf{X}_2 = \mathbf{R}_{21}^S\mathbf{X}_1 + \mathbf{T}_{21} \tag{4.29}$$

Therefore, the rotation and translation of the relative orientation is related to the transformation in the global coordinate system as [122]:

$$\mathbf{R}_{21}^S = \mathbf{R}_2^T\mathbf{R}_1 \tag{4.30}$$

$$\mathbf{T}_{21} = \mathbf{R}_2^T(\mathbf{T}_1 - \mathbf{T}_2) \tag{4.31}$$

Here, $\mathbf{R}_{21}^S$, is a rotation matrix with small angle approximation as in Eq. (4.5), $\mathbf{R}_2$ and $\mathbf{R_1}$ are full 3D rotation matrices as given in Eq. (4.3). This leads to the following equation:

$$\mathbf{R}_{21}^S = \begin{pmatrix} c\varphi_2 c\kappa_2 & -c\varphi_2 s\kappa_2 & s\varphi_2 \\ c\omega_2 s\kappa_2 + s\omega_2 s\varphi_2 c\kappa_2 & c\omega_2 c\kappa_2 - s\omega_2 s\varphi_2 s\kappa_2 & -s\omega_2 c\varphi_2 \\ s\omega_2 s\kappa_2 - c\omega_2 s\varphi_2 c\kappa_2 & s\omega_2 c\kappa_2 + c\omega_2 s\varphi_2 s\kappa_2 & c\omega_2 c\varphi_2 \end{pmatrix}^T \begin{pmatrix} c\varphi_1 c\kappa_1 & -c\varphi_1 s\kappa_1 & s\varphi_1 \\ c\omega_1 s\kappa_1 + s\omega_1 s\varphi_1 c\kappa_1 & c\omega_1 c\kappa_1 - s\omega_1 s\varphi_1 s\kappa_1 & -s\omega_1 c\varphi_1 \\ s\omega_1 s\kappa_1 - c\omega_1 s\varphi_1 c\kappa_1 & s\omega_1 c\kappa_1 + c\omega_1 s\varphi_1 s\kappa_1 & c\omega_1 c\varphi_1 \end{pmatrix} \tag{4.32}$$

The matrix $\mathbf{R}_{21}^S$ is skew symmetric (Eq. (4.5)), however in the Eq. (4.30) the multiplication of the rotation matrices $\mathbf{R}_2^T\mathbf{R}_1$ does not result in a skew symmetric matrix. For having a skew symmetric matrix on both sides of Eq. (4.30), subtract the transpose of Eq. (4.30) to itself:

$$\mathbf{R}_{21}^S - {\mathbf{R}_{21}^S}^T = {\mathbf{R}_2}^T\mathbf{R}_1 - {\mathbf{R}_1}^T\mathbf{R}_2 \tag{4.33}$$

$$2\mathbf{R}_{21}^{S} = \mathbf{R}_{2}^{T}\mathbf{R}_{1} - \mathbf{R}_{1}^{T}\mathbf{R}_{2} \tag{4.34}$$

This gives the relationship between angles (R_X, R_Y, R_Z) and angles $(\omega_1, \varphi_1, \kappa_1, \omega_2, \varphi_2, \kappa_2)$. The angles (R_X, R_Y, R_Z) are not directly measured, but are estimated using the method given in Section 4.1 (Eqs. (4.19) and (4.21)). These estimated values along with their estimated variance can be used as observations in the bundle adjustment. Similarly, the relative translation between the two images (T_X, T_Y, T_Z) is related to the absolute orientation parameters as in Eq. (4.31). Using Eqs. (4.34) and (4.31) and applying small angle approximations e.g. $\lim_{\omega\to 0} cos(\omega) = 1$ and $\lim_{\omega\to 0} sin(\omega) = \omega$, following observation equations can be written:

$$\begin{aligned} R_X = &\quad 0.5 * ((c\varphi_2 c\kappa_1 + c\varphi_1 c\kappa_2)(\omega_2 - \omega_1) + (\varphi_2 - \varphi_1)(s\kappa_1 + s\kappa_2)) \\ R_Y = &\quad 0.5 * ((c\varphi_2 s\kappa_1 + c\varphi_1 s\kappa_2)(\omega_1 - \omega_2) + (\varphi_2 - \varphi_1)(c\kappa_1 + c\kappa_2)) \\ R_Z = &\quad 0.5 * (2(\kappa_2 - \kappa_1) + (\omega_2 - \omega_1)(s\varphi_1 + s\varphi_2)) \end{aligned} \tag{4.35}$$

$$\begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} = \begin{bmatrix} c\varphi_2 c\kappa_2 & c\omega_2 s\kappa_2 + s\omega_2 s\varphi_2 c\kappa_2 & s\omega_2 s\kappa_2 - c\omega_2 s\varphi_2 c\kappa_2 \\ -c\varphi_2 s\kappa_2 & c\omega_2 c\kappa_2 - s\omega_2 s\varphi_2 s\kappa_2 & s\omega_2 c\kappa_2 + c\omega_2 s\varphi_2 s\kappa_2 \\ s\varphi_2 & -s\omega_2 c\varphi_2 & c\omega_2 c\varphi_2 \end{bmatrix} \begin{bmatrix} T_{2X} - T_{1X} \\ T_{2Y} - T_{1Y} \\ T_{2Z} - T_{1Z} \end{bmatrix} \tag{4.36}$$

Eqs. (4.35) and (4.36) shows the relationship between the parameters of exterior orientation of two frames and relative orientation between these two frames. Hence, given a relative orientation of an image pair, six equations can be written in the 12 unknowns of exterior orientation of each image in the pair. The weighting of each of these equations is done according to the *a posteriori* covariance matrix of the relative orientation result. Here, it should be mentioned that the coefficients for some of the terms in Eq. (4.19) are similar which may lead to high correlation between the estimated parameters. Therefore, the result of estimated relative orientation should be interpreted along with the covariance of the estimated parameters. For example, a small rotation R_X, results in a similar motion as the translation T_Y [3, 29, 142]. Consequently, the correlation coefficient can be higher among these parameters. This ambiguity in rotation and translation is more pronounced if the field of view is small [8, 100]. In the least squares estimation of the relative orientation, the diagonal elements of 6×6 covariance matrix gives the variance of each estimated parameter and the off diagonal elements determines the correlation between the parameters. Therefore, it is important to use the full 6×6 covariance matrix (Eq. (4.22)) for weighting of Eqs. (4.36) and (4.35) in bundle adjustment.

Bundle adjustment used in this work now has three different types of observation equations i.e. collinearity equations Eq. (2.3), depth observations Eq. (4.26) and estimated relative orientation (Eqs. (4.36) and (4.35)). As they are heterogeneous group of observations, therefore they are weighted according to their *a priori* variances. For example, accuracy of feature matching can be half a pixel and the accuracy of distance measurement can be 1 cm. The weighting of relative orientation equations is done according to covariance matrix obtained from Eq. (4.22). Using variance component analysis the *a priori* variances can be adjusted or refined for weighting of the corresponding observation equations in the next iteration of bundle adjustment. The procedure of variance component analysis as followed from [139] is briefly described now. The

$\mathbf{Q}$ matrix given in Eq. (4.22) is the covariance matrix of the observations. As there are three groups of observations, the full covariance matrix $\mathbf{Q}_{Full}$ can be decomposed into individual parts:

$$\mathbf{Q}_{Full} = \sigma^2_{FeatMatch}\mathbf{Q}_1 + \sigma^2_{DistObs}\mathbf{Q}_2 + \sigma^2_{RelOrient}\mathbf{Q}_3 \tag{4.37}$$

Here $\sigma^2_{FeatMatch}, \sigma^2_{DistObs}, \sigma^2_{RelOrient}$ are the variances of groups of observations comprising feature matching, distance observations and estimated relative orientations respectively. While $\mathbf{Q}_1, \mathbf{Q}_2$ and $\mathbf{Q}_3$ are the non overlapping matrices comprising of cofactors for each group of observations. The adjustment is started with *a priori* approximates of the variances of each group. In the next adjustment the *a posteriori* variances of each group of observations are used for weighting of each group of observations. This process is continued until the *a priori* weights are equal to the *a posteriori* weights. More information on this procedure can be found in the relevant texts [107, 139].

The three groups of equations used here in bundle adjustment are all non linear, therefore, they are first linearized at the approximate values of the unknown parameters, and then iteratively updated based on the updated parameter values until the solution converges to a minimum. For a large number of images the number of observation equations can be very large, and thus resulting in a large system of equations. However, the coefficient matrix $\mathbf{A}$ in bundle adjustment is largely sparse and this sparsity is utilized in computing efficient solution for bundle adjustment [108, 190]. The covariance matrix for the orientation parameters and the 3D points can be computed similar to Eq. (4.22) [64].

CHAPTER 5

Motion of Independently Moving Objects

In this chapter, a method of estimating motion of independently moving objects in image sequences with a static camera is presented. The goal is to estimate dense 3D motion vectors for the entire image, which may contain multiple independently moving objects. In 2D this is a typical optical flow problem. In this work, the focus is on full 3D motion estimation as the depth information is available along with the intensity images. Therefore, for each pixel, three components of the motion are estimated. The estimation of these components are realized using a two step algorithm, where the first step is the local motion estimation and the second step is the global regularization. The first step is similar to Lucas Kanade [117] type optical flow estimation which may result in partially dense flow fields as only local information is utilized. In the second step the flow vectors and the corresponding accuracy estimates are used in a global regularization procedure to obtain dense smooth motion vectors. The advantage of this two step procedure is that it leads to a simpler formulation and the resulting equation systems can be solved by robust least squares adjustment, in both the steps. Many recent optical flow algorithms optimize an energy function, which includes brightness constancy assumption, smoothness constraint and occlusion detection together with robust estimation. However, the optimization of a non convex energy function of data with higher noise levels (the depth data from range sensors is typically more noisy then color images), while estimating 3D motion vectors instead of 2D motion, is non trivial. Furthermore, the quality measures for the least squares adjustment are well defined, using these measures the accuracy and uncertainty of the flow vectors can be estimated and exploited accordingly in the smoothness or the regularization procedure as given in the next sections.

5.1 Local Motion Estimation

The proposed local motion estimation algorithm is based on integrating the range flow and optical flow constraints similar to the method of relative orientation presented in the previous Chap-

ter 4. Thus, depth and intensity information are exploited simultaneously. As the motion in videos is under consideration it is assumed that motion is small due to high temporal sampling. First the 3D motion constraint based on range flow is derived. The range flow constraint in terms of derivatives in image space is given as in Eq. (2.21).

$$\dot{Z} = Z_x\dot{x} + Z_y\dot{y} + Z_t.$$

Here, again assuming a pinhole perspective projection model and assuming that the global coordinate system is aligned with the sensor coordinate system, the relation between the image motion and the object motion is given by:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \frac{1}{Z}\begin{bmatrix}f & 0 & -x\\ 0 & f & -y\end{bmatrix}\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{bmatrix}. \tag{5.1}$$

Z is the depth, x and y are the image coordinates. f is the principal distance. Substituting the pixel velocities in Eq. (2.21) by Eq. (5.1) results in:

$$\begin{bmatrix} Z_x\frac{f}{Z} & Z_y\frac{f}{Z} & -\frac{Z_x x + Z_y y}{Z} - 1\end{bmatrix}\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{bmatrix} = \begin{bmatrix}-Z_t\end{bmatrix}. \tag{5.2}$$

Eq. (5.2) gives the relation between the 3D velocity $(\dot{X}, \dot{Y}, \dot{Z})$ of the point, and the spatio temporal derivatives of depth. This 3D velocity is given in the coordinate system attached with the camera. For each pixel in the image, one such constraint can be written.

Similarly, the optical flow can be used for intensity images to derive a constraint for 3D motion using intensity and the depth information. As described previously the optical flow constraint equation (or brightness constancy assumption) is given by:

$$I_x\dot{x} + I_y\dot{y} = -I_t. \tag{5.3}$$

Here, I_x, I_y and I_t are the spatial and temporal derivatives of the intensities, respectively. As the depth of the object point is available, the pixel velocities in Eq. (5.1) are substituted in Eq. (5.3) to obtain a modified optical flow constraint [78], which contains full 3D motion components:

$$\begin{bmatrix} I_x\frac{f}{Z} & I_y\frac{f}{Z} & -\frac{I_x x + I_y y}{Z}\end{bmatrix}\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{bmatrix} = \begin{bmatrix}-I_t\end{bmatrix}. \tag{5.4}$$

The unknowns $(\dot{X}, \dot{Y}, \dot{Z})$ in Eqs. (5.2) and (5.4) are the components of 3D object velocity of each pixel that are computed using the spatial and temporal derivatives of depth and intensity. For each pixel, there are two constraints for three unknown velocity components. Now, similar to Lucas Kanade optical flow it is assumed that motion in the local neighborhood is similar and a square window is selected around the subject pixel. For a window of $n \times n$ pixels, $2n^2$ constraints or equations are available and this overdetermined system of equations can be solved

by least squares adjustment. The selection of window size is important as a smaller window size may not contain enough information (aperture problem) and a larger window size may contain multiple motions, thus invalidating the homogeneous motion assumption. The system of equations (Eqs. (5.2) and (5.4)) can be written in the form

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{l} + \mathbf{e}. \tag{5.5}$$

where $\boldsymbol{\beta}$ are the unknowns $(\dot{X}, \dot{Y}, \dot{Z})^\top$. The observations $\mathbf{l}$ contain the change in depth and intensity per pixel and $\mathbf{e}$ is the residual vector. In this work, an ordinary least squares (OLS) solution is used to estimate the parameters. The solution of the system of equations is given by:

$$\boldsymbol{\beta} = (\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{l}. \tag{5.6}$$

Here, $\mathbf{Q}$ is a diagonal matrix which contains the *a priori* variances of the observations [133]. The inverse of $\mathbf{Q}$ acts as weights of observations in the adjustment. The solution in Eq. (5.6) uses an inverse of $\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}$ (or $\mathbf{A}^T\mathbf{A}$ if $\mathbf{Q}$ is identity matrix), which encodes the texture in intensity and geometry. The three eigenvalues of the $\mathbf{A}^T\mathbf{A}$ matrix (also known as structure tensor) can be used to analyze the amount of texture in the pixel neighborhood. If there is enough texture present in the image or if there is enough variation in the surface normals e.g. as in a corner (three intersecting planes), then all three eigenvalues are greater than zero and all three components of the velocity can be determined. In presence of only an edge, one eigenvalue will be close to zero and the motion normal to the edge can be determined. In case of a planar structure with homogeneous gray level, two eigenvalues of $\mathbf{A}^T\mathbf{A}$ will be close to zero and only the motion direction perpendicular to the plane can be estimated [177]. If any eigenvalue of this matrix is zero then this matrix is singular and cannot be inverted and the solution has to be computed using matrix pseudo inverse [102]. However, in most cases due to presence of noise, the eigenvalues are not exactly zero and the matrix is invertible although it may be badly conditioned.

The variance of the estimated unknowns is given by:

$$\mathbf{Q}_\beta = \sigma_0^2(\mathbf{A}^\mathbf{T}\mathbf{Q}^{-\mathbf{1}}\mathbf{A})^{-\mathbf{1}}, \tag{5.7}$$

$$\sigma_0^2 = \frac{\mathbf{e}^T\mathbf{Q}^{-1}\mathbf{e}}{n-3}, \quad \mathbf{e} = \mathbf{A}\boldsymbol{\beta} - \mathbf{y}, \tag{5.8}$$

where n is the number of observations. The diagonal elements of matrix $\mathbf{Q}_\beta$ give an estimate of the precision of each parameter ($\boldsymbol{\beta} = [\dot{X}, \dot{Y}, \dot{Z}]^T$). As written above, the components of $\mathbf{Q}_\beta$ indicate how accurately the individual motion components can be estimated depending on the amount of texture available. The σ_0 value gives the measure of overall quality of the adjustment. In presence of multiple motions and occlusions σ_0 is large [177]. Consequently, the resulting values in the diagonal elements of $\mathbf{Q}_\beta$ are also large. Therefore, these quality measures are essential in interpretation of the estimated flow field and indicates the failure of the assumptions or the functional model. Shadowing, occlusions and scattering can cause optical flow and range flow constraint to become invalid. Therefore, the pixels with inaccurate flow vectors should be filtered out. Thus, the diagonal components of $\mathbf{Q}_\beta$ are used as a quality measure for the corresponding estimated flow vector components [177]. In particular, the flow vectors that exceed a fixed threshold σ_{max}^2 in the diagonal elements of $\mathbf{Q}_\beta$ are removed. In contrast, components

or vectors of the flow that were computed with high precision (i.e., that are below the fixed threshold σ^2_{max}) are kept. Thus, the local motion estimation may result in a partially dense flow vectors.

5.2 Estimation Models for Optical Flow and Range Flow

As mentioned before, both range flow and optical flow are under-constrained for a single pixel as there are more unknowns than the flow constraints. The common strategy is to estimate flow using flow constraints over a neighborhood, this gives an over determined linear system of equations which can be solved using the least squares method. Different parameter estimation models for solving this equation system can be used e.g. OLS or Gauss Markov model [102,117, 120] (see Eq. (5.5) above), total least squares [175, 177], constrained total least squares [191] and Gauss Helmert model [70].

The optical flow constraint Eq. (5.3) and range flow Eq. (2.19) constraint can be written in matrix form as:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = -I_t. \tag{5.9}$$

$$\begin{bmatrix} Z_X & Z_Y & 1 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = -Z_t, \tag{5.10}$$

This over-determined system of equations can be written in the form

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{l}, \tag{5.11}$$

here, $\mathbf{A}$ is the matrix of coefficients, $\boldsymbol{\beta}$ is the unknown motion vector components and $\mathbf{l}$ are the observations. If OLS or Gauss Markov model is used for parameter estimation of the form Eq. (5.11), it is assumed that only observations ($\mathbf{l}$) contains error of stochastic nature and the coefficients of $\mathbf{A}$ are exact. In reality the derivatives of intensity or depth are also computed using measurements corrupted with noise, so they also contain some error. However, it is expected that the effect of neglecting these errors is not significant in the Gauss Markov model if the noise in the two images is uncorrelated and normally distributed [5, 60]. Therefore, in this work, Gauss Markov model has been used, which has the standard form:

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{l} + \mathbf{e}, \tag{5.12}$$

where $\mathbf{e}$ is the error in measurement. The solution of the Gauss Markov model or ordinary least squares, given earlier in Eq. (5.6) is obtained by minimizing $\mathbf{e}^T\mathbf{e}$.

In comparison to OLS, the method of Total Least Squares (TotalLS) [161, 194], can model the errors in the coefficient matrix as well. The general form of TotalLS for equation system of the form Eq. (5.11) can be written as:

$$(\mathbf{A} - \mathbf{E}_\mathbf{A})\boldsymbol{\beta} = \mathbf{l} + \mathbf{e}, \tag{5.13}$$

where $\mathbf{E}_A$ contains the stochastic errors in the coefficient matrix and $\mathbf{e}$ is the error vector for the observations $\mathbf{l}$. The TotalLS solution for Eq. 5.14 minimizes the following sum:

$$\mathbf{e}^T\mathbf{e} + {\mathbf{e}_A}^T\mathbf{e}_A \tag{5.14}$$

where, $\mathbf{e}_A = vect(\mathbf{E}_A)$ is the columns of $\mathbf{E}_A$ stacked into one column vector. The solution of TotalLS (Eq. (5.14)) can be computed using singular value decomposition or eigen decomposition. A simple example which shows the difference between OLS and TotalLS is 2D line fitting. An OLS solution minimizes the vertical distances between the points and the estimated line, while the TotalLS solution minimizes the orthogonal distances between the points and the estimated line [136, 160].

Spies et al. [175, 177] have used TotalLS for estimating range flow. The range flow constraint:

$$Z_X\dot{X} + Z_Y\dot{Y} - \dot{Z} + Z_t = 0, \tag{5.15}$$

can be written in the form,

$$\mathbf{A}\boldsymbol{\beta} = 0, \tag{5.16}$$

and solved as an eigenvalue problem. Here, $\mathbf{A} = \begin{bmatrix} Z_X & Z_Y & -1 & Z_t \end{bmatrix}$ and $\boldsymbol{\beta} = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} & 1 \end{bmatrix}$. From the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$, range flow is computed as [177]:

$$\boldsymbol{\beta} = \frac{\mathbf{1}}{ev_4}\begin{bmatrix} ev_1 \\ ev_2 \\ ev_3 \end{bmatrix} \tag{5.17}$$

Spies et al. [175, 177] have also presented solutions to compute range flow vectors for cases with planar and linear geometry using eigenvectors of $\mathbf{A}^T\mathbf{A}$.

Based on the standard TotalLS formulation given in Eq. (5.14), the matrix $\mathbf{E}_A$ contains error value for each term of matrix $\mathbf{A}$. However, the coefficient of $\dot{Z}$ is always -1 in Eq. (5.15) and is thus error free. Therefore, TotalLS solution based on Eq. (5.17), will cause bias in the estimation of the unknowns as it can add an error (as given in general formulation of TotalLS in Eq. (5.14)) to coefficient of $\dot{Z}$ which is error free. To resolve this issue Garbe et al. [47] proposed a solution using mixed OLS and TotalLS which doesn't assign error to exactly known coefficient of $\dot{Z}$. The general TotalLS solution as also used in [175, 177] makes the assumption that the errors in the Z_X,Z_Y and Z_t are uncorrelated. This is however, not true because the derivatives are computed over a pixel neighborhood and this induces correlation between these terms. Therefore, it is suspected that this may lead to a biased parameter estimation. Therefore different TotalLS based solution which can take into account correlation between the terms needs to be investigated [6, 173].

As compared to range flow, there is a larger collection of literature in optical flow, investigating error modeling of the optical flow constraint [134, 138, 191, 202]. In optical flow as well, TotalLS has been used to model errors in spatial derivatives along with the time derivatives of intensity [200, 202], similar to TotalLS approach in range flow. As mentioned previously, without taking into account the correlation between the terms of the flow equation, TotalLS may give a biased solution. As the standard form of TotalLS don't model this correlation, different solutions have been proposed to estimate flow by modeling the errors and the correlations in both

spatio-temporal derivatives [134, 138, 191]. The Gauss Helmert model [70] also known as the general case of least squares adjustment can also be used for modeling errors and correlations in the spatial and temporal derivatives of the intensity and depth.

5.3 Global Regularization

The goal of the regularization step is to estimate a complete 3D motion for each pixel. In this section the reference to previously (Section 5.1) estimated flow vectors ($\boldsymbol{\beta}$), is made using the pixels indexes (i, j) (e.g., $\boldsymbol{\beta}_{i,j}$). The global regularization step integrates the information from the set of all flow vectors from the local motion estimation step and smoothness prior by minimizing the following energy function [77, 177]:

$$E_{reg} = E_{data} + E_{sm}. \tag{5.18}$$

The term E_{data} uses the previously estimated flow vectors $\boldsymbol{\beta}_{i,j}$ at pixel coordinates (i, j) and the corresponding confidence values $\mathbf{Q}_\beta$ (the pixel's subscripts (i, j) are not used with $\mathbf{Q}_\beta$) from the local motion estimation to minimize the following sum:

$$E_{data} = \sum_{i,j} \{(\mathbf{v}_{i,j} - \boldsymbol{\beta}_{i,j})^T \mathbf{Q}_\beta^{-1} (\mathbf{v}_{i,j} - \boldsymbol{\beta}_{i,j})\}. \tag{5.19}$$

Here, $\mathbf{v}_{i,j}$ denotes a regularized unknown flow vector. The data term ensures that the difference $\mathbf{v} - \boldsymbol{\beta}$ is low. This is especially true for flow vectors $\boldsymbol{\beta}$ that were computed with high accuracy. The correlation between the neighboring flow vectors, arising due to overlapping area of the local neighborhood is neglected.

The second term in the regularization scheme, the smoothness term E_{sm}, assumes similar motion among neighboring pixels. Consequently, it minimizes the sum of differences of neighboring 3D velocities [182]:

$$E_{sm} = \sum_{i,j} (\mathbf{v}_{i,j} - \mathbf{v}_{i+1,j})^T \mathbf{P}_{s,\Delta i} (\mathbf{v}_{i,j} - \mathbf{v}_{i+1,j}) + (\mathbf{v}_{i,j} - \mathbf{v}_{i,j+1})^T \mathbf{P}_{s,\Delta j} (\mathbf{v}_{i,j} - \mathbf{v}_{i,j+1}) \tag{5.20}$$

To regularize the previously estimated flow vectors, we perform a global least squares estimation that minimizes both terms (i.e., Eqs. (5.19) and (5.20)) over the entire image. In Eq. (5.18) no relative weighting of the terms E_{data} and E_{sm} is specified, because the respective accuracy of the observations provides the weighting. It is given in the matrices $\mathbf{Q}_\beta^{-1}$ and $\mathbf{P}_{s,\Delta i}$, $\mathbf{P}_{s,\Delta j}$, respectively, which are defined below. The first set of observation equations, which represent Eq. (5.19), can be written in the form:

$$\mathbf{v}_{i,j} = \boldsymbol{\beta}_{i,j} + \mathbf{e}_{i,j,\beta}. \tag{5.21}$$

As stated above, the precision $\sigma^2_{\beta_{i,j}}$ is derived during the local flow estimation (i.e., Eq. (5.7)) and corresponds to the variance of each estimated velocity component in $\mathbf{Q}_\beta$. This is equal to a weight of $1/\sigma^2_{\beta_{i,j}}$.

The observation equations for the smoothness term (i.e., Eq. (5.20)) can be written in the form:

$$\mathbf{v}_{i,j} - \mathbf{v}_{i+1,j} = \mathbf{0} + \mathbf{e}_{i,j,\Delta i}, \tag{5.22}$$

$$\mathbf{v}_{i,j} - \mathbf{v}_{i,j+1} = \mathbf{0} + \mathbf{e}_{i,j,\Delta j}. \tag{5.23}$$

The weight $P_{s,\Delta i}$ for each observation equation in Eq. (5.22) is computed by:

$$P_{s,\Delta i} = \left(\frac{1}{\sigma_s^2}\right) g_I(|I_{i,j} - I_{i+1,j}|) g_Z(|Z_{i,j} - Z_{i+1,j}|), \tag{5.24}$$

$$\sigma_s^2 = \sigma_{max}^2 - max(\sigma_{\beta_{i,j}}^2, \sigma_{\beta_{i+1,j}}^2). \tag{5.25}$$

$P_{s,\Delta j}$ is defined analogously. In Eq. (5.24), g_I and g_Z are weighting functions that are based on intensity and depth differences of the corresponding pixels, respectively. This weighting causes an anisotropic behavior of the smoothness term by reducing the influence of the smoothness term across depth or intensity gradients [206]. Here, Gaussian functions are used for g_I and g_Z as given in Eq. (5.26), where σ_Z and σ_I are empirically chosen.

$$\begin{aligned} g_Z(Z) &= exp(-\tfrac{(Z_{i,j} - Z_{i+1,j})^2}{2{\sigma_Z}^2}) \\ g_I(I) &= exp(-\tfrac{(I_{i,j} - I_{i+1,j})^2}{2{\sigma_I}^2}) \end{aligned} \tag{5.26}$$

Furthermore, Eq. (5.24) considers the precision of the estimation of each individual flow vector from the previous step [177]. σ_{max}^2 is the threshold that corresponds to the largest variance allowed in the local motion estimation. If either of the two flow vectors at locations (i, j) or $(i+1, j)$ was determined with low accuracy in the first motion estimation step, the corresponding smoothness observation obtains a large weight. Weighting of the remaining smoothness equations is performed analogously. The regularization step computes an iteratively re-weighted least squares solution to reduce the influence of outliers.

By approximating $\mathbf{Q}_\beta$ as a diagonal matrix (neglecting the off-diagonal elements), the equation system (Eq. (5.19)) can be split into $\dot{X}$, $\dot{Y}$ and $\dot{Z}$ components, which are independent of each other. This allows a faster computation. Furthermore, the equation system (Eqs. (5.21),(5.22) and (5.23)) is linear, but due to robust outlier detection, iterations still need to be performed. Furthermore, the equations system is large as three observation equations are written for each component of velocity at each pixel. However, this equation system is largely sparse which is utilized to solve this system as a linear least squares estimation problem.

CHAPTER 6

Experiments

In this chapter, the qualitative and quantitative evaluation of the methods presented in Chapters 4 and 5 is given. Here, data from different types of range sensors presented in Chapter 3 is used. Evaluating the methods on different types of data sets is important in assessing the applicability and accuracy of the proposed methods. The experimental results are subdivided into sections camera motion, motion of independently moving objects and motion estimation over a landslide. The results are further compared to ground truth (GT) data (when available) for the quantitative evaluation.

6.1 Camera Motion

Relative Orientation

In this section, the evaluation results of the relative orientation method presented in Chapter 4 are presented. First the results on a ToF camera are presented, which is then followed by results on the RGB-D SLAM dataset and benchmark. Furthermore, the robustness of the algorithm in presence of outliers is evaluated by estimating camera motion in presence of an independently moving object.

To weight the depth and intensity based constraints in Eq. (4.19) of relative orientation method, two options were analyzed. In the first approach, the depth and the intensity images were normalized to have same mean spatial derivatives for use in Eq. (4.19), similar to approach of Spies et al. [178]. While in the second approach, the weighting of the depth and intensity terms was done by using the noise models for intensity and depth observations. The noise models for intensity observations of range camera and Kinect were derived empirically. The accuracy model of Khoshelham and Elberink [99] was used for weighting the depth observations of Kinect and Eq. (3.5) was used for weighting the range observations from the range camera. However, during the analysis no significant differences in the results were observed, when weighting the intensity and depth constraints in Eqs. (4.19) using the two approaches. The results given here are computed using the first approach, i.e. by normalizing the intensity and depth images.

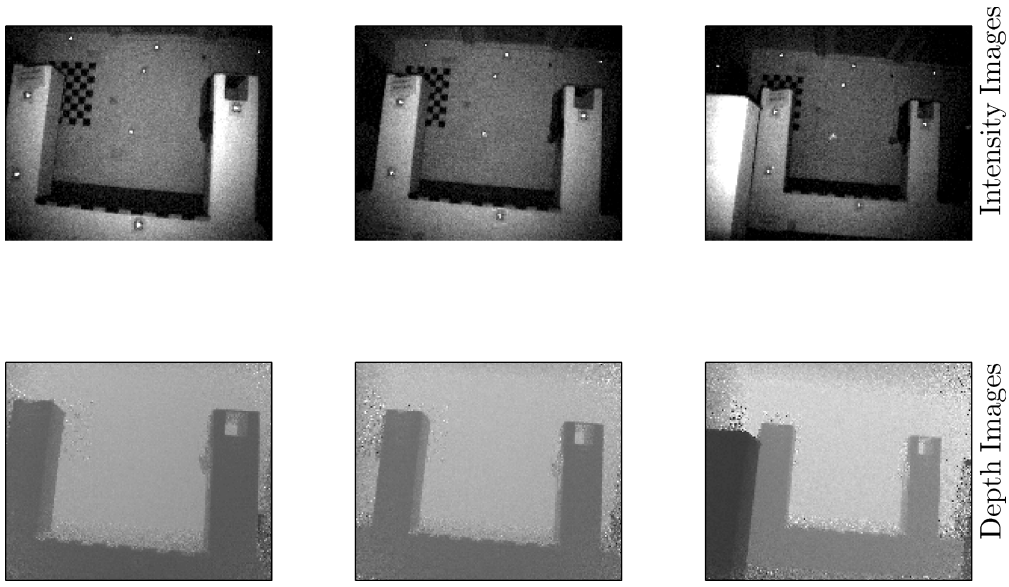

Figure 6.1: Three Frames from the video of a static scene with a moving camera. Frame 30 (left), Frame 160 (middle) and Frame 340 (right)

The ToF sequences are captured with an SR3000 camera. The calibration presented in [94] is applied to remove systematic effects in the range observations. To compare the results of the algorithm with a reference data, ground control points are included in the scene and the software package ORPHEUS [93] is used to perform bundle adjustment on the tracked ground control points in the sequences. The results from ORPHEUS are used as Ground Truth (GT) for quantitative analyses of the algorithm.

In the first type of experiments, the camera moves in a static environment. Figure 6.1 shows three intensity and range images from one of the sequence (involving only camera motion). Figure 6.2 shows the estimated camera trajectory in $X - Z$ plane of the camera coordinate system relative to the first frame in the video sequence consisting of approx. 350 frames. The magnitude of the translation corresponds well with the GT, however the accumulation of errors results in the drift and the estimated motion deviates more with time. In a second example of a static scene, the focus was set on a dominant rotational component of the camera motion (a ϕ rotation around the Y-axis of the camera). Similar results to the first scene are observed and the accumulation of the error in the ϕ rotation are visualized in Figure 6.3.

In the second type of experiments, the scene includes an independently moving object, therefore the observed motion has two contributions, one from the camera motion and the other from the independently moving object. For dealing with independently moving object, the pixels occupied by the object need to be excluded in the computation of the camera's motion. This can be achieved by using robust adjustment, during which a weighting function is applied to iteratively reduce the weight of the outliers (i.e. pixels of the independently moving object). Figure 6.4 shows the weighting of the optical flow and range flow based terms for each pixel (black corresponds to low weight). The dominant motion is due to the camera movement. Therefore, the

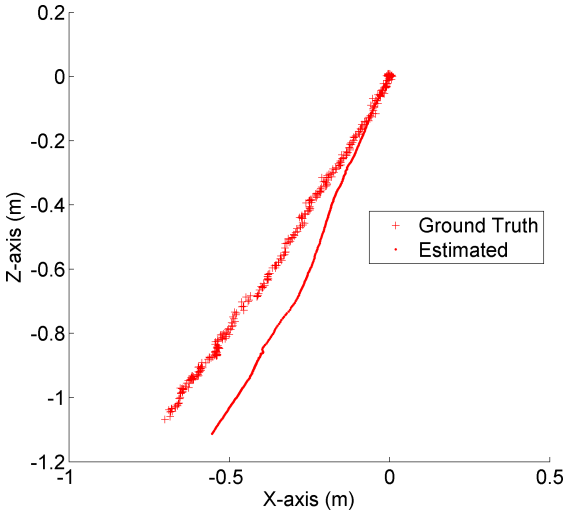

Figure 6.2: First static scene: Comparison of the trajectory from the GT

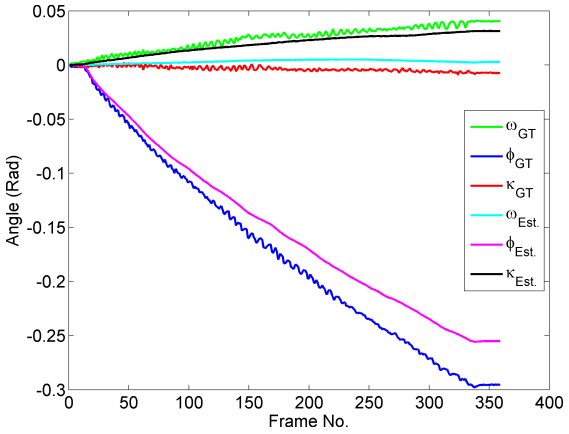

Figure 6.3: Second static scene: Camera rotation and comparison with the GT

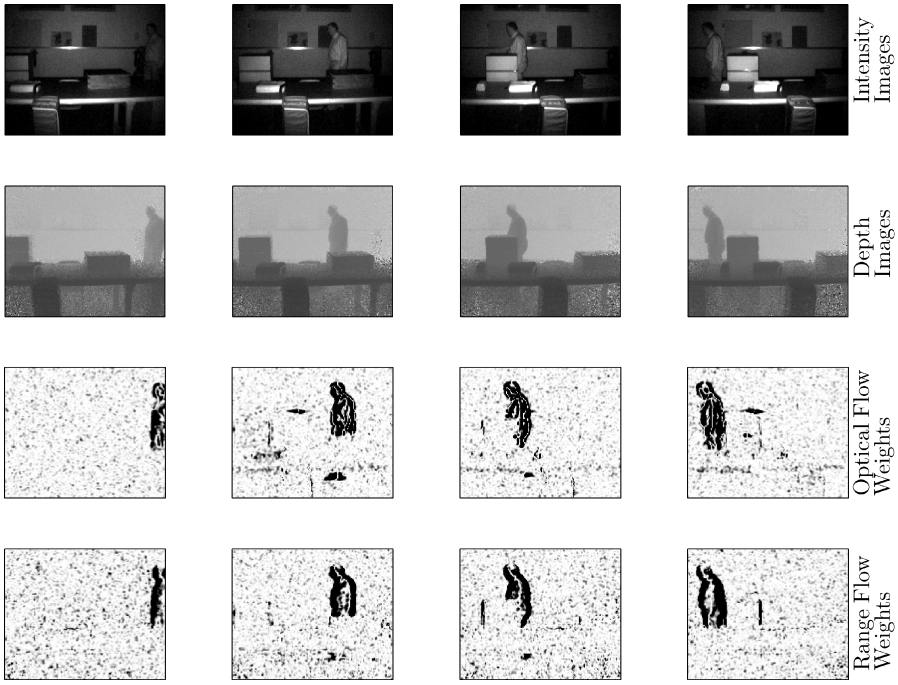

Figure 6.4: Camera motion and an independently moving object. Darker gray tone represents lower weights

static parts of the scene receive high weight in the robust adjustment, while the pixels belonging to the independently moving object get a low weight. However, the robust estimation is sensitive to the number of outliers. Therefore, in the presence of large number of outliers (multiple independent moving objects or moving object covering large parts of the scene) the breakdown point of the robust estimators can be reached. In this experiment, reference camera motion was measured with a linear scale bar and the motion has a magnitude of $95\mathrm{cm}$. The difference between the estimated and the referenced camera motion is around $5\mathrm{cm}$.

The relative orientation method is further tested on the RGB-D SLAM data set and benchmark [180], which contains numerous videos of indoor environment captured with Microsoft Kinect and Asus Xtion RGB-D cameras. The dataset contains a variety of scenes, some of them are recorded with hand held camera motion while some scenes are captured by a camera attached to a moving robot. The ground truth camera trajectory was obtained using a high accuracy motion capture system consisting of eight high speed tracking cameras. From the calibration of the motion capture system, it was concluded that the relative error on consecutive frames is lower than 1 mm and 0.5 degrees. Additionally, the absolute error over the entire motion capture area is lower than 10 mm and 0.5 degrees.

In addition to the ground truth trajectory the results from the RGB-D SLAM algorithm [39] are also available. This RGBD-SLAM algorithm [39] consists of the following steps. First distinctive feature are extracted and matched in the color images, which gives the 3D point correspondences between two frames as depth of these points are available in the depth images.

Based on these point correspondences the relative transformation between two frames is estimated using RANSAC strategy. These relative poses are then refined using a modified ICP algorithm [165]. The globally consistent poses are then estimated using the HOGMAN pose graph optimization [58].

In this dataset most of the sequences contain relatively large motion (observed image motion is up to about 20 pixels) between consecutive frames. The rotational movement is upto 50 deg/sec. As a result motion blur is quite significant when the camera undergoes fast motion. Furthermore, Kinect selects the exposure time automatically, as a consequence significant illumination changes are also present in between the consecutive frames. Hence, motion blur and illumination changes along with limited texture and geometry as is common in a typical indoor environment, poses a challenging task for camera motion estimation and makes this dataset a suitable platform for evaluation of methods presented in this thesis for estimation of camera motion.

The evaluation of the estimated camera motion is performed by computing the following two measures: The Relative Pose Error (RPE) and the Absolute Trajectory Error (ATE). RPE is a suitable measure for evaluation of the local accuracy of camera motion. Therefore, the relative orientation method presented in Section 4.1 is evaluated using RPE. On the other hand ATE is suited for measuring the overall accuracy or the global consistency of the camera trajectory. Therefore, the bundle adjustment with relative orientation (Section 4.2) is evaluated using ATE. If $\{\mathbf{Q}_1, \mathbf{Q}_2, ...\mathbf{Q}_n\}$ is the set of estimated camera poses or camera trajectory and $\{\mathbf{G}_1, \mathbf{G}_2, ...\mathbf{G}_n\}$ is the ground truth trajectory consisting of n camera poses then RPE for time instant i is computed as:

$$\mathbf{RPE}_i = (\mathbf{Q}_i^{-1}\mathbf{Q}_{i+1})^{-1}(\mathbf{G}_i^{-1}\mathbf{G}_{i+1}) \tag{6.1}$$

The poses $\mathbf{Q}_i$ and $\mathbf{G}_i$ are represented as 4×4 matrices:

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{T}_i \\ \mathbf{0} & 1 \end{bmatrix} \tag{6.2}$$

where, $\mathbf{R}_i$ is a 3×3 rotation matrix representing the angular attitude of the camera and $\mathbf{T}_i$ is a three dimensional vector representing the position of the projection center of the i^{th} frame with reference to the common coordinate system. If there are n camera poses, then the root mean squared error (RMSE) for the translational components is computed as:

$$RMSE(\mathbf{RPE}_{1:n-1}) = \left(\frac{1}{n-1} \sum_{i=1}^{n-1} \|trans(\mathbf{RPE})_i\|^2 \right)^{\frac{1}{2}} \tag{6.3}$$

where, $trans(\mathbf{RPE})_i$ contains the translational components of the i^{th} relative pose error. The RMSE error for the rotational components is computed similarly.

As the camera trajectories can be given in any arbitrary coordinate system, to compute ATE, the trajectories are first aligned using closed form solution of Horn [79]. If $\mathbf{S}$ is the transformation that aligns estimated trajectory to the ground truth trajectory then ATE at time instant i is computed as:

Table 6.1: Translational (millimeters) part of **RPE** between consecutive frames on several sequences of the RGB-D Dataset [180].

		Relative Orientation Translational Error (mm)			**RGBD SLAM Translational Error (mm)**		
Sequence	**Frames**	**RMSE**	**Mean**	**Median**	**RMSE**	**Mean**	**Median**
FR1 xyz	798	4.9	4.0	3.2	5.7	4.8	4.1
FR1 rpy	722	6.8	5.2	3.9	12.1	8.4	5.6
FR1 desk	595	6.8	5.4	4.2	11.7	8.3	5.9
FR1 desk2	639	7.0	5.6	4.6	17.5	9.9	6.4
FR2 xyz	3665	2.1	1.7	1.4	2.0	1.7	1.5

Table 6.2: Rotational (degrees) part of **RPE** between consecutive frames on several sequences of the RGB-D Dataset [180].

		Relative Orientation Rotational Error (deg)			**RGBD SLAM Rotational Error (deg)**		
Sequence	**Frames**	**RMSE**	**Mean**	**Median**	**RMSE**	**Mean**	**Median**
FR1 xyz	798	0.39	0.32	0.25	0.35	0.30	0.23
FR1 rpy	722	0.82	0.70	0.65	0.91	0.64	0.47
FR1 desk	595	0.76	0.64	0.56	0.73	0.49	0.34
FR1 desk2	639	0.75	0.65	0.58	1.0	0.61	0.39
FR2 xyz	3665	0.23	0.19	0.16	0.21	0.17	0.11

$$\mathbf{ATE}_i = \mathbf{G}_i^{-1}\mathbf{S}\mathbf{Q}_i \tag{6.4}$$

$$RMSE(\mathbf{ATE}_{i:n}) = \left(\frac{1}{n}\sum_{i=1}^{n}\|trans(\mathbf{ATE})_i\|^2\right)^{\frac{1}{2}} \tag{6.5}$$

where, $trans(\mathbf{ATE})_i$ contains the translational components of the i^{th} absolute trajectory error.

In the RGB-D SLAM database and benchmark [180], an online tool and script is provided for computation of RPE and ATE, which also performs the alignment of the trajectories for computation of ATE. It also provides the error statistics of the whole trajectory.

The RPE (Eq. (6.1)) is computed for each consecutive pair of frames for several sequences in this dataset which is given in Tables 6.1 and 6.2. The RMSE, mean and median translational errors of relative orientation parameters between consecutive frames of the sequences are in Table 6.1, while the errors for the rotational terms are given in Table 6.2. These results were computing using the relative orientation method within a coarse to fine framework due to presence of large motion. The color images were converted to gray scale. The missing data values in the depth images were interpolated and an iterative re-weighted adjustment was performed to remove the outliers. All the relative orientations are transformed into the coordinate system of the first camera frame, this gives all camera poses with reference to the first camera position.

Figure 6.5: RGB-D SLAM dataset [180] sample images from several sequences

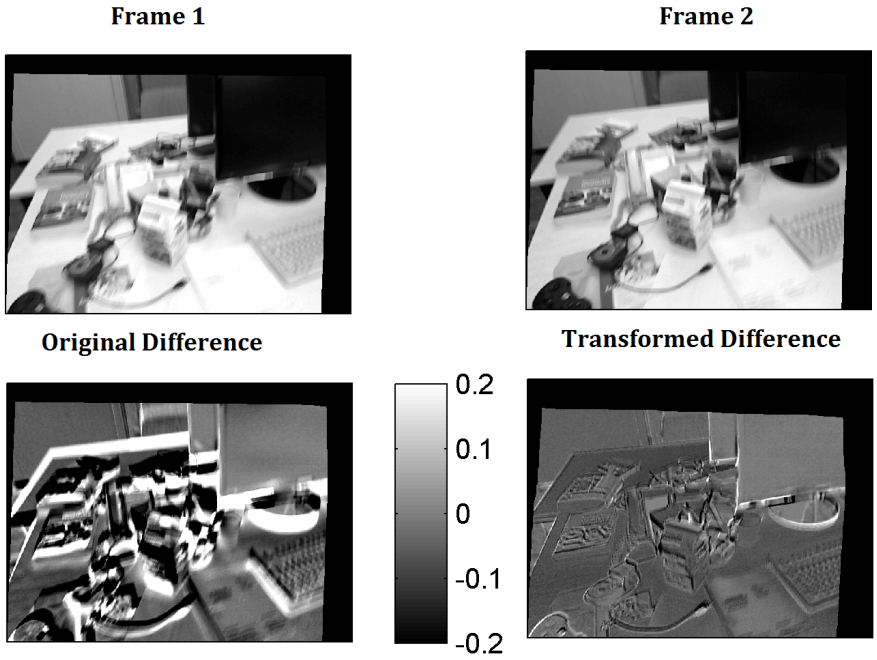

Figure 6.6: Top: Two frames from *FR1 desk* sequence. 2nd Row Left: Original difference between the two frames. 2nd Row Right: Difference between two images after transforming the 2nd image using relative orientation parameters. The darker gray levels on homogeneous areas of the bottom right image indicates illumination changes. Images are normalized to [0-1]. Motion blur and changes in illumination is quite evident.

The sequence *FR1 xyz* and *FR 2 xyz* contains mainly translational motions, *FR1 rpy* contains rotation along the 3 axes, *FR1 desk* and *FR1 desk2* contains both rotational and translational movements in a typical office environment (Figure 6.5). The evaluation results on these different sequences show good accuracy compared to the RGB-D SLAM algorithm [39]. The rotational error is less than 1 degree for all the five sequences and the median translational error is in the order of few millimeters. Thus, the good performance of the algorithm on these variety of scenes shows the validity of the proposed method. These sequences contains fast translational and rotational movements involving different types of indoor scenes. The accuracy of the results show that even image motion greater than 20 pixels per frame is not a problem as the coarse to fine strategy is able to handle motions of such a magnitude. Furthermore, due to high redundancy and robust estimation, the relative orientation is accurately estimated even with motion blur and illumination changes as shown in Figure 6.6.

Bundle Adjustment with Relative Orientation

As discussed before, the relative orientation method doesn't produce a globally consistent solution as the errors accumulate along the trajectory. The RPE presented in Tables 6.1 and 6.2

Table 6.3: Absolute Trajectory Error (ATE) for relative orientations method compared to the RGB-D SLAM [39].

	Relative Orientation Trajectory ATE (cm)			**RGB-D SLAM Trajectory ATE (cm)**		
Sequence	**RMSE**	**Mean**	**Median**	**RMSE**	**Mean**	**Median**
FR1 xyz	2.75	2.57	2.47	1.34	1.20	1.11
FR1 rpy	10.1	9.7	9.24	2.87	2.45	2.15
FR1 desk	5.70	5.41	5.34	2.58	2.31	2.13
FR1 desk2	11.7	10.5	10.1	4.2	3.5	3.1
FR2 xyz	6.4	5.6	4.9	2.6	2.2	2.0

evaluates only the relative accuracy of the camera poses. The global consistency of the trajectory is measured by computing the ATE. In Table 6.3 the ATE of the trajectory computed from the relative orientation is given along with the trajectory from RGB-D SLAM [39] algorithm. Clearly the ATE error from relative orientation is much higher compared to the RGB-D SLAM [39]. The accumulation of the error is also visible in Figure 6.7, which shows the trajectories from relative orientation and RGB-D SLAM along with the ground truth for sequences *FR1 xyz, FR1 rpy, FR1 desk, FR1 desk2, FR2 xyz* in the X, Y plane of the motion capture system.

To obtain globally consistent camera trajectory, bundle adjustment using the relative orientation constraints is performed over these sequences. First the SURF features are extracted and matched among images to find corresponding points in images. As there are usually false point matches, RANSAC is used to find a set of inlier points. Here, a minimum of 15 inlier points satisfying a projective transformation are used for a successful matching of an image pair. At this point, it is essential to point out that due to high motion blur combined with low texture, enough image features could not be matched for some of the frames. Therefore, for some frames no features points are used in the bundle adjustment. However, as the relative orientation parameters of these frames are available, the exterior orientation of these frames is still optimized in the bundle adjustment. This fact also emphasizes the reason behind using the relative orientations as observations in the bundle adjustment.

As the bundle adjustment is a non linear optimization, it requires an approximate solution. The approximate solution can be computed by estimating the transformation between 3D points (as depth is also available for matched features points) using e.g. the closed form solution as given in Chapter 2. Another option is to use the relative orientations transformed into a common coordinate system for the approximate solution. During the analysis, it is observed that the relative orientation solution provided a better approximation for initialization of bundle adjustment for these sequences, which is also evident from Figure 6.7, as the trajectories correspond well to the ground truth. Having good approximate values of the unknowns can lead to faster convergence and therefore, less number of iterations needs to be performed in bundle adjustment.

The bundle adjustment used here, consists of three different types of observations, it is essential to assign correct weights for each group of observations, which is done by choosing *a priori* variances for each group of observations. The accuracy of the feature matching is typically in sub pixel range $[0.5 - 1]$ pixel, therefore the accuracy of feature matching is selected

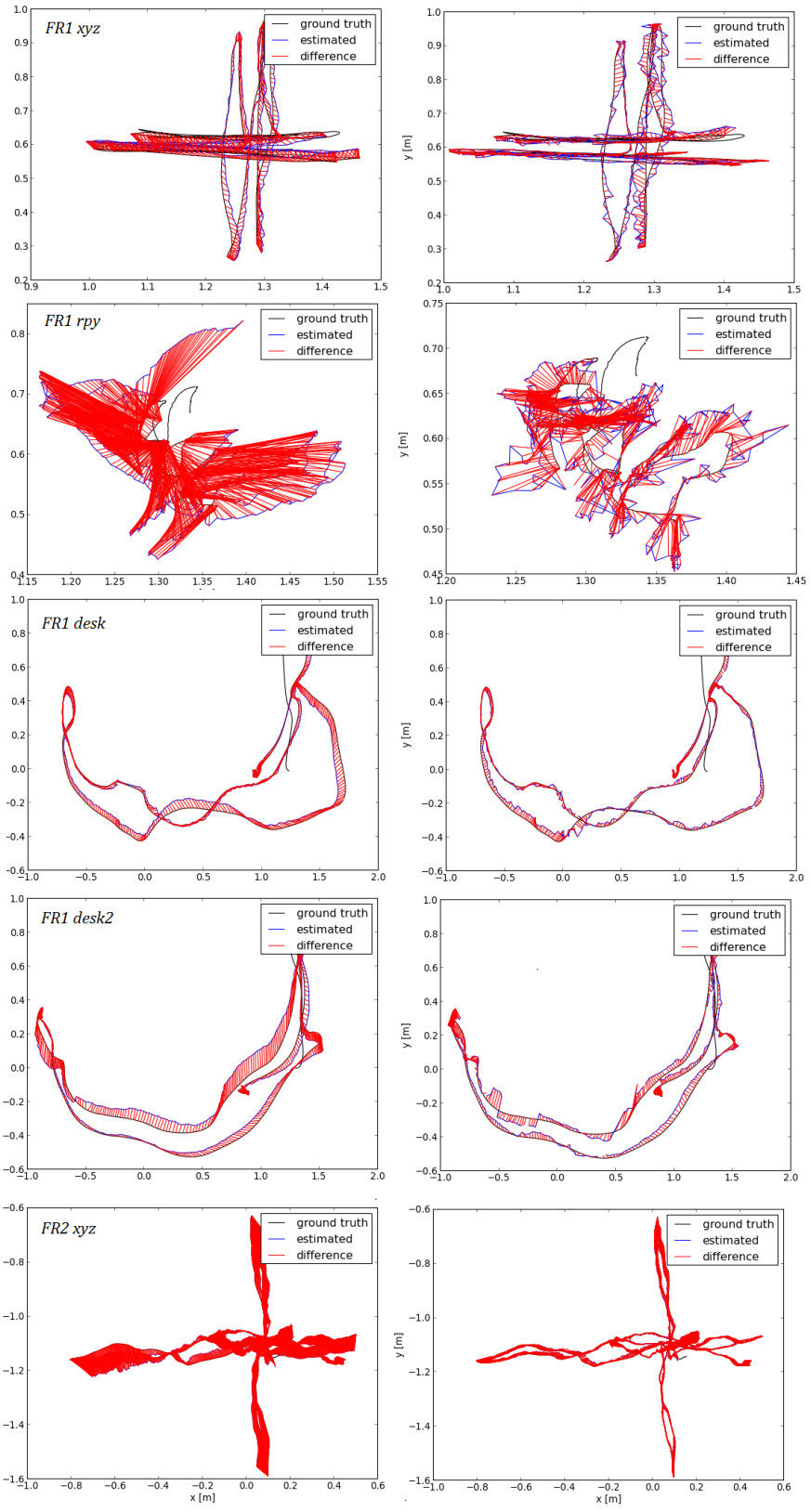

Figure 6.7: Camera trajectories estimated using relative orientation (left) and RGB-D SLAM (right)

Table 6.4: Variance factors of the three observation groups

Sequence	FR1 xyz	FR1 rpy	FR1 desk	FR1 desk2	FR2 xyz
$\sigma^2_{0,FeatMatch}$	0.62	0.82	0.86	0.58	0.3
$\sigma^2_{0,DepthObs}$	8.83	13.5	15	19	8.4
$\sigma^2_{0,RelOrient}$	3.9e5	2.1e6	5.0e5	6.6e5	3.4e5

as 0.9 (pixel). For the accuracy of depth measurements, the model given in Khoshelham and Elberink [99] can be used, which says that the standard deviation of the depth error increases quadratically from a couple of millimeters at 0.5 m depth to around 4 cm at 5 m depth. Khoshelham and Elberink [99] verified the model by experimentally observing the residuals of plane fitting at different depths from the camera. The weighting of the observations corresponding to the estimated relative orientations deserves some attention. The *a posteriori* covariance matrix $\mathbf{Q}_\beta$ of the unknowns in Eq. (4.22) gives an estimate of the accuracy of the relative orientation parameters. This covariance matrix can be used for weighting of the observation equations corresponding to the estimated relative orientations. As an example, the $\mathbf{Q}_\beta$ matrix computed from relative orientation estimation of an image pair shown in Figure 6.8 is:

$$\mathbf{Q}_\beta = 1e^{-10} \begin{bmatrix} 0.1567 & -0.0111 & -0.0229 & 0.0034 & 0.1158 & -0.0127 \\ -0.0111 & 0.0867 & 0.0136 & -0.0646 & -0.0074 & 0.0028 \\ -0.0229 & 0.0136 & 0.0843 & -0.0062 & -0.0224 & 0.0110 \\ 0.0034 & -0.0646 & -0.0062 & 0.0530 & 0.0009 & -0.0032 \\ 0.1158 & -0.0074 & -0.0224 & 0.0009 & 0.0926 & -0.0088 \\ -0.0127 & 0.0028 & 0.0110 & -0.0032 & -0.0088 & 0.0343 \end{bmatrix} \tag{6.6}$$

The standard deviation of the three components of translation is $1e^{-6}[3.9, 2.9, 2.9]$ (meters). This estimate of the accuracy of relative orientation parameters is highly optimistic. One reason for this highly optimistic estimate is the redundancy number which is used for computing σ_0 (Eq. (4.23)) and subsequently $\mathbf{Q}_\beta$ (Eq. (4.22)). If all the pixels in the depth and intensity image are used for estimation of relative orientation (Eq. (4.19)) then the redundancy is $(2 \times 640 \times 480) - 6 = 614,394$. In reality however, a number of observation equations corresponding to diminishing intensity and depth gradients provide little or weak constraints in determining the unknown parameters, but these observation equations still add up to the redundancy number. Furthermore, as discussed in Section 5.2, while using ordinary least squares it is assumed that the coefficients are free of error but this is an assumption, which also leads to an optimistic estimate of the accuracy of the unknowns. Therefore, the covariance estimates from relative orientation may only give a biased accuracy estimates of the relative orientation observations.

To obtain better estimates of accuracy of each group of observations, variance component analysis is performed [139]. In this procedure the *a posteriori* variance of each group of observations is computed after each adjustment and then these estimates are used as *a priori* variance estimates for the next iteration of bundle adjustment. Table 6.4 shows the variance factors for the three observations groups after few iterations. The cofactor matrix in Eq. (4.37) for feature points is an identity matrix, while the cofactor matrix for depth observations comprises the depth

accuracy model from Khoshelham and Elberink [99] and the cofactor matrix for the relative orientation comprises of covariance matrices (Eq. (4.22)) obtained from the relative orientation estimation.

Now using the $\sigma^2_{0,RelOrient}$ of the *FR1 desk* as the variance estimate for the observations equations of relative orientation group, the co-variance matrix given in Eq. (6.6) multiplied by $\sigma^2_{0,RelOrient}$ gives the following estimate of variance of the relative orientation parameters of image pair shown in Figure 6.8:

$$\mathbf{Q}_{\beta} = 1e^{-5} \begin{bmatrix} 0.7835 & -0.0555 & -0.1145 & 0.0170 & 0.5790 & -0.0635 \\ -0.0555 & 0.4335 & 0.0680 & -0.3230 & -0.0370 & 0.0140 \\ -0.1145 & 0.0680 & 0.4215 & -0.0310 & -0.1120 & 0.0550 \\ 0.0170 & -0.3230 & -0.0310 & 0.2650 & 0.0045 & -0.0160 \\ 0.5790 & -0.0370 & -0.1120 & 0.0045 & 0.4630 & -0.0440 \\ -0.0635 & 0.0140 & 0.0550 & -0.0160 & -0.0440 & 0.1715 \end{bmatrix} \tag{6.7}$$

Thus, $[0.0028, 0.0021, 0.0021]$ (meters) is the estimated standard deviation of the three translational components of the relative orientation for the image pair shown in Figure 6.8, after using variance factors given in Table 6.4. These accuracy estimates corresponds well to the $\mathbf{RPE}$ given in Table 6.1. This shows that the accuracy estimates from the *a posteriori* covariance matrix of relative orientation are indeed too optimistic. Figure 6.9 shows the difference in the trajectories when using the original covariance matrix from relative orientation and when using the revised covariance estimate from variance component analysis. It is clear that in the latter case the estimated trajectory corresponds better to the ground truth trajectory. Furthermore, the trajectories from relative orientation and bundle adjustment with relative orientation using original covariance matrix are quite similar. This is due to the fact that the covariance matrix as give in Eq. (6.6) leads to very high weights for the relative orientation observations and therefore, the bundle adjustment solution will tend to correspond to these observations as it will assign less error to these observation equations due to their higher weighting.

The variance factor for depth observations in Table 6.4 shows that the accuracy model of Khoshelham and Elberink [99] is also optimistic. This may be due to the fact that the evaluation of the model was done in a controlled environment, while, the sequences from the RGB-D dataset are more complex in terms of the scene structure and the camera motion. In fact, it is observed that assigning all the depth observations an equal weight instead of using the model of [99] improves the accuracy of the trajectories.

The result of bundle adjustment with relative orientation is shown in Figure 6.10 and in Table 6.5. The accuracy of the estimated trajectory is even better than RGB-D SLAM [39] algorithm for several sequences. This shows that the proposed algorithm performs well on challenging sequences. The Table 6.6, shows the mean standard deviation of the estimated projection centers from the bundle adjustment solution. In comparison to the ATE given in Table 6.5, these accuracy estimates are optimistic. One reason for the optimistic estimate is that the estimated accuracy of the unknowns in the bundle adjustment, shows how well the observations *fit* to each other and this estimated may not be well representative of the absolute trajectory of the camera.

The computational aspects of the method are now briefly discussed. The method of relative orientation is computationally inexpensive, the coefficients of $\mathbf{A}$ matrix in Eq. (4.19) only

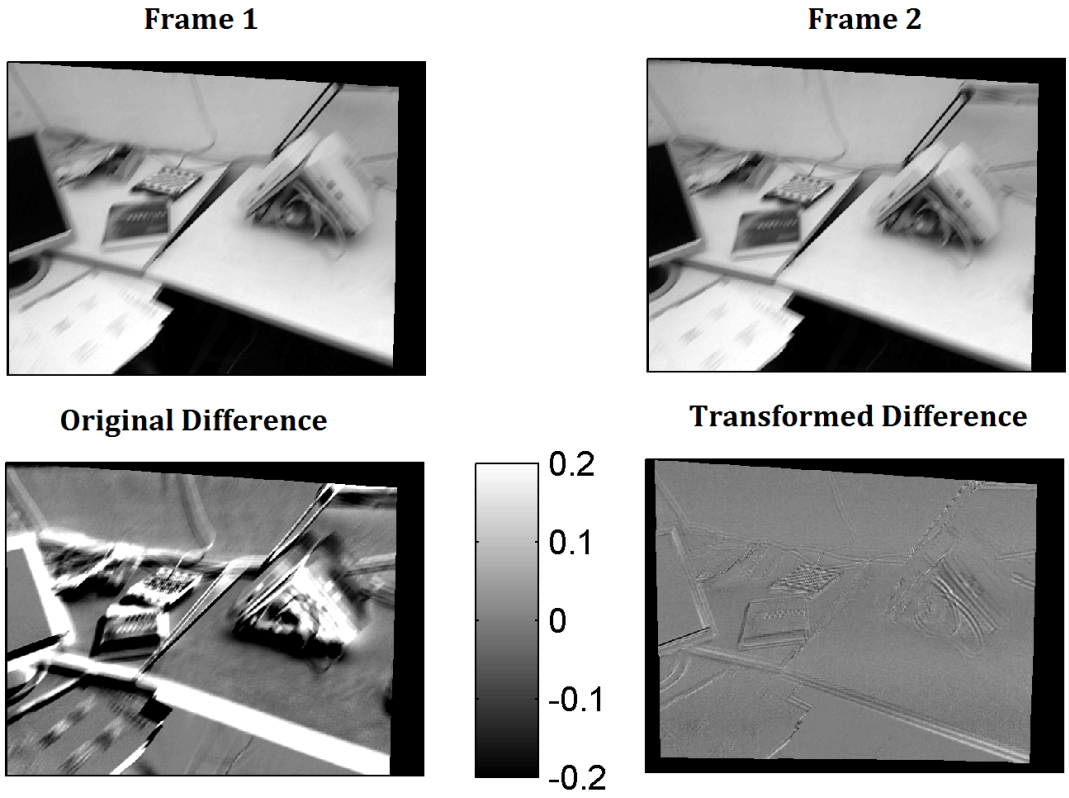

Figure 6.8: Top: Two frames from *FR1 desk* sequence. 2nd Row Left: Original difference between the two frames. 2nd Row Right: Difference between two images after transforming the 2nd image using relative orientation parameters. Images are normalized to [0-1].

Table 6.5: Absolute Trajectory Error (ATE) for bundle adjustment with relative orientation compared to the RGB-D SLAM [39].

	Bundle Adjustment with Relative Orientation ATE (cm)			**RGB-D SLAM Trajectory ATE (cm)**		
Sequence	**RMSE**	**Mean**	**Median**	**RMSE**	**Mean**	**Median**
FR1 xyz	1.34	1.19	1.05	1.34	1.20	1.11
FR1 rpy	1.9	1.75	1.6	2.87	2.45	2.15
FR1 desk	2.8	2.33	2.0	2.58	2.31	2.13
FR1 desk2	4.0	3.3	2.9	4.2	3.5	3.1
FR2 xyz	1.5	1.3	1.2	2.6	2.2	2.0

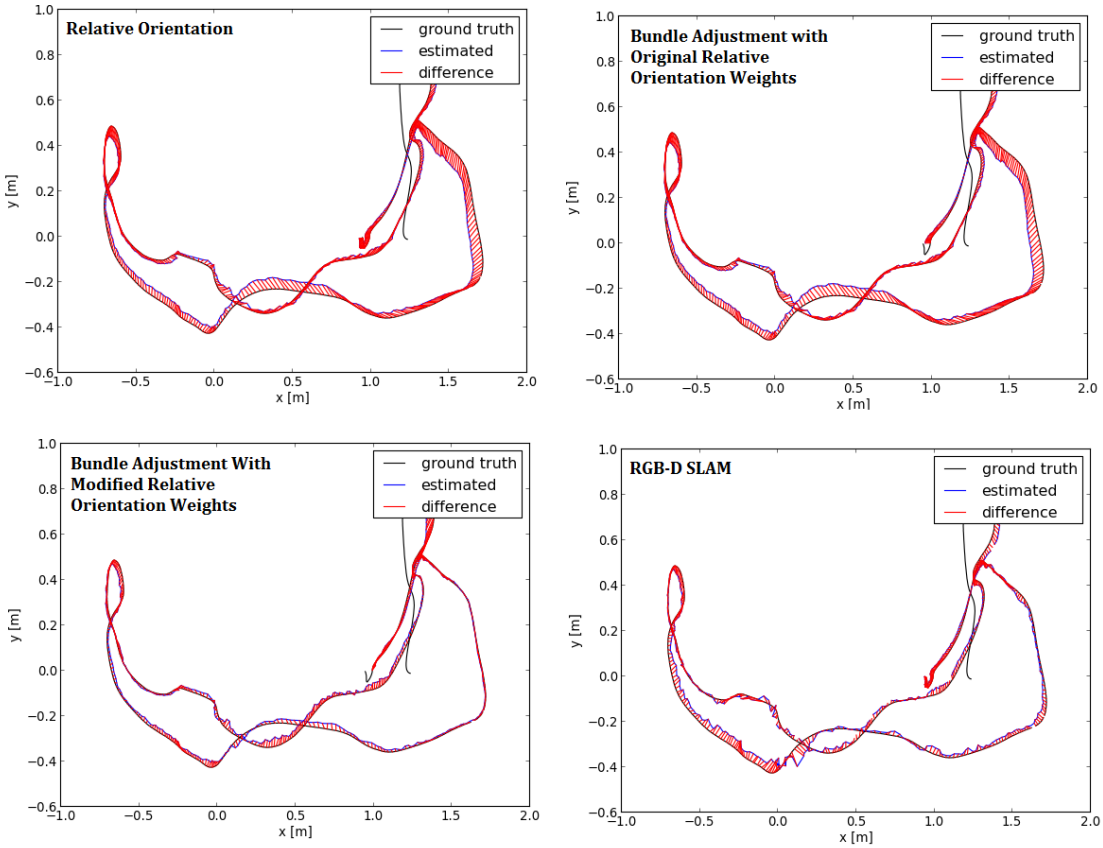

Figure 6.9: Comparison of the trajectory of *FR1 desk* sequence, computed using *a)* Only relative orientation *b)* Bundle adjustment with relative orientation constraints using the original covariance matrix from relative orientation for weighting of the relative orientation observations. *c)* Bundle adjustment with relative orientation constraints but using weighting of the relative orientation terms using variance component analysis *d)* Trajectory from RGB-D SLAM [39]

Table 6.6: Mean standard deviation (in cm) of the projection center of each frame from bundle adjustment

Sequence	FR1 xyz	FR1 rpy	FR1 desk	FR1 desk2	FR2 xyz
Projection centers (mean std. dev).	0.26	0.44	0.8	0.9	0.15

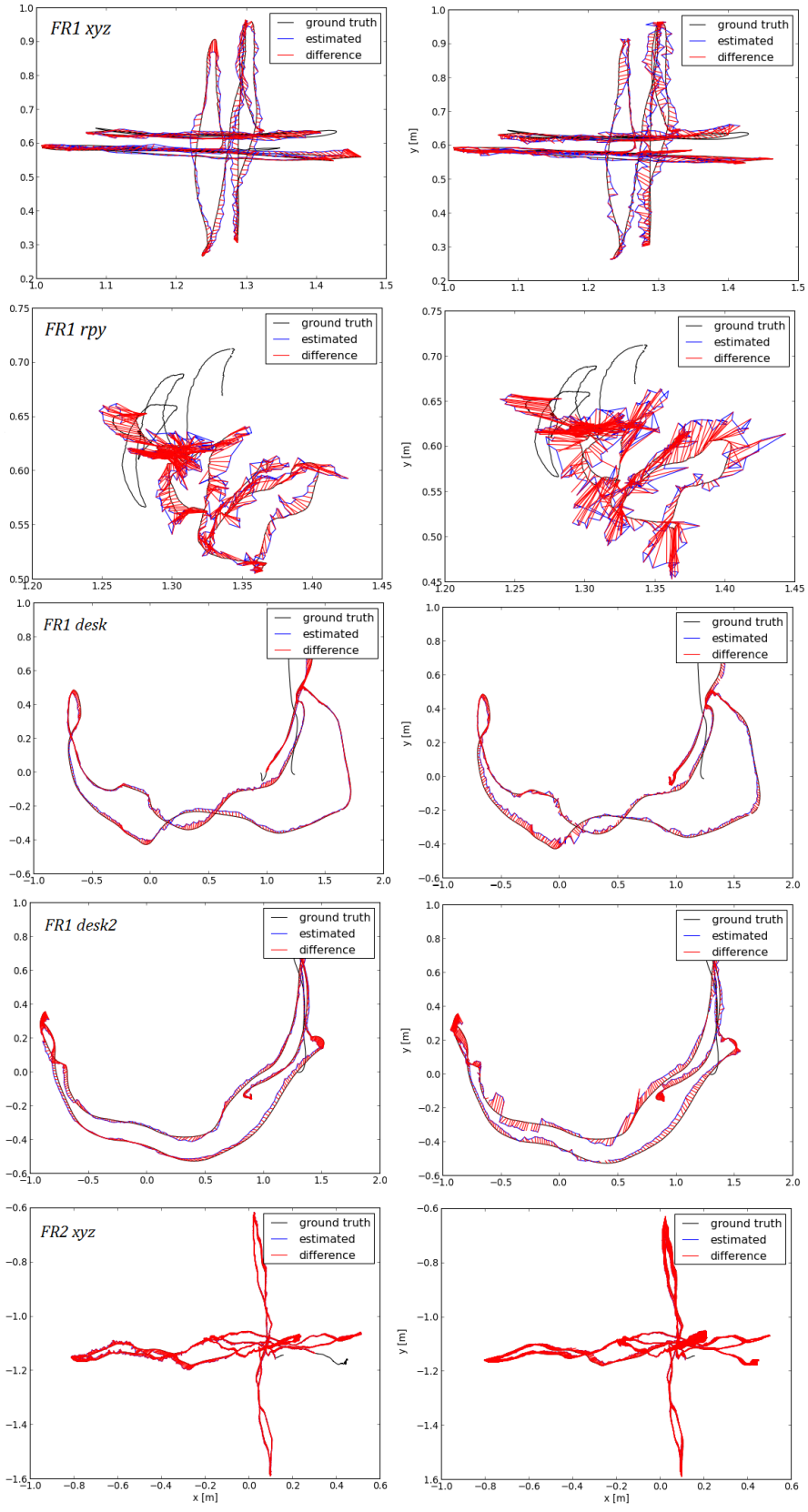

Figure 6.10: Camera trajectories estimated using bundle adjustment with relative orientation (left) and RGB-D SLAM (right)

requires computation of the intensity and range derivatives, which can be computed very efficiently. The least squares adjustment involves only six unknowns, therefore, the adjustment computation is also inexpensive. The bundle adjustment and variance component analysis on the other hand can become computationally expensive as the number of images and number of features increase. The state of the art SLAM algorithms for large scale mapping, focus on optimizing only large number of poses without performing bundle adjustment using feature matching. Therefore, if a large scale mapping is desired, then the number of features points should be kept limited, or only the relative orientations between frames should be optimized. Another strategy, which is often adopted is to perform a local bundle adjustment i.e. to optimize the orientations and 3D point locations using a subset of images, and repeat this procedure for all images of the sequence.

6.2 Motion of Independently Moving Objects

In this section, the method for motion estimation of independently moving object is evaluated. Three sequences: one synthetic scene *cubes* (Figure 6.11) and two real world scenes *trains* (Figure 6.15) and *people* (Figure 6.16) are used to evaluate the proposed method. The real world scenes are captured using the an SR3000 ToF camera and to compensate for the systematic distortions, calibration presented in [94] has been applied. Another real world test scene not presented here is given in [49].

The quantitative analysis of first two sequences (*cubes* and *trains*) is performed by computing Angular Error (AE) and the Endpoint Error (EE). The AE measures the difference in the direction of the motion vectors and is computed as:

$$AE = cos^{-1}\left(\frac{\dot{X}\times\dot{X}_{GT}+\dot{Y}\times\dot{Y}_{GT}+\dot{Z}\times\dot{Z}_{GT}}{\sqrt{\dot{X}^2+\dot{Y}^2+\dot{Z}^2}\sqrt{\dot{X}_{GT}^2+\dot{Y}_{GT}^2+\dot{Z}_{GT}^2}}\right) \tag{6.8}$$

where, $(\dot{X}_{GT}, \dot{Y}_{GT}, \dot{Z}_{GT})$ is the ground truth motion vector and $(\dot{X}, \dot{Y}, \dot{Z})$ is the estimated 3D motion vector. The EE measures the difference in the end points of the estimated motion vector and the ground truth motion vector.

$$EE = \sqrt{(\dot{X}-\dot{X}_{GT})^2+(\dot{Y}-\dot{Y}_{GT})^2+(\dot{Z}-\dot{Z}_{GT})^2} \tag{6.9}$$

The AE and EE are commonly used error measures for evaluation of optical flow algorithms [9, 10]. For the third sequence *people*, no ground truth is available so only qualitative analysis is performed.

Synthetic Scene

In the synthetic scene (*cubes*) two cubes are moving on a ground plane in front of a wall (Figure 6.11). It has a resolution of 201×161 pixels. The scene's lateral extent is 18 meters. The depth ranges from 5 to 18 meters. The GT flow vector for the cube in the front is $[0.07, 0, 0.01]$, while for cube in the back the GT motion is $[-0.14, 0, 0]$ (meters per frame).

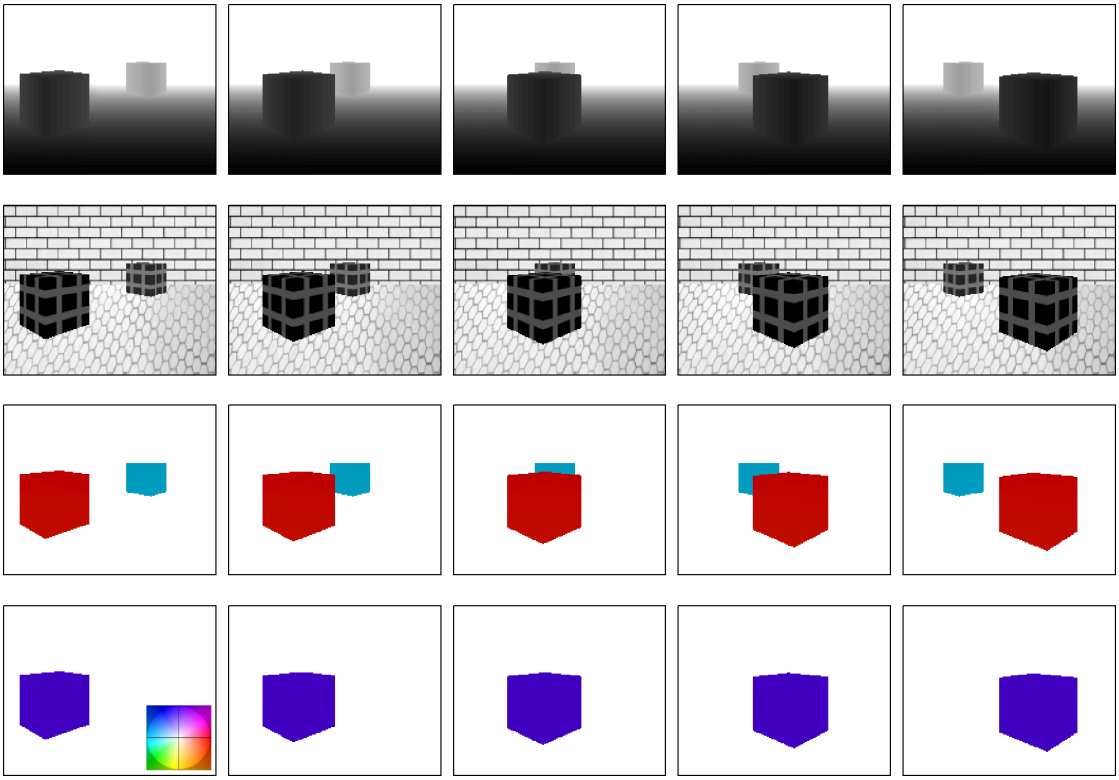

Figure 6.11: *1st Row*: Depth images for five frames of the sequence. *2nd Row* Corresponding intensity images. *3rd Row*: $(\dot{X}, \dot{Y})$ components of estimated motion vectors. *4th Row*: $\dot{Z}$ component of the estimated motion vectors. The motion vectors are color coded (color wheel) as: Hue encodes orientation and saturation encodes magnitude. $\dot{Z}$ is represented by only vertical axis of the color wheel.

The results of the proposed motion estimation on *cubes* are shown in Figure 6.11. It can be seen that the estimated flow vectors are visually accurate, even when more than 60 percent of the background cube is occluded (e.g., background cube in the middle frame, Figure 6.11). Furthermore, the boundaries of the objects are well determined. Figure 6.11 also demonstrates that the global regularization scheme (i.e., the weighting of the smoothness term based on depth and intensity differences) effectively avoids over-smoothing. Even object boundaries at critical locations, such as at low incidence angles (e.g., top surface of cube) and similar depths (e.g., contact points of ground plane and cube) are well identified. The visual quality of the results (Figure 6.11) are confirmed by the quantitative error measures. The average EE and the average AE for this entire sequence are 0.1 millimeters and 0.1 degrees, respectively.

The steps of the algorithms are illustrated in Figure 6.12. The results of the local motion estimation are shown in the left column. It can be seen that towards the borders of the moving cubes, the flow vectors are erroneous as the local neighborhood contains both moving and static objects. This is depicted in the standard deviation of the estimated flow vectors, shown in the

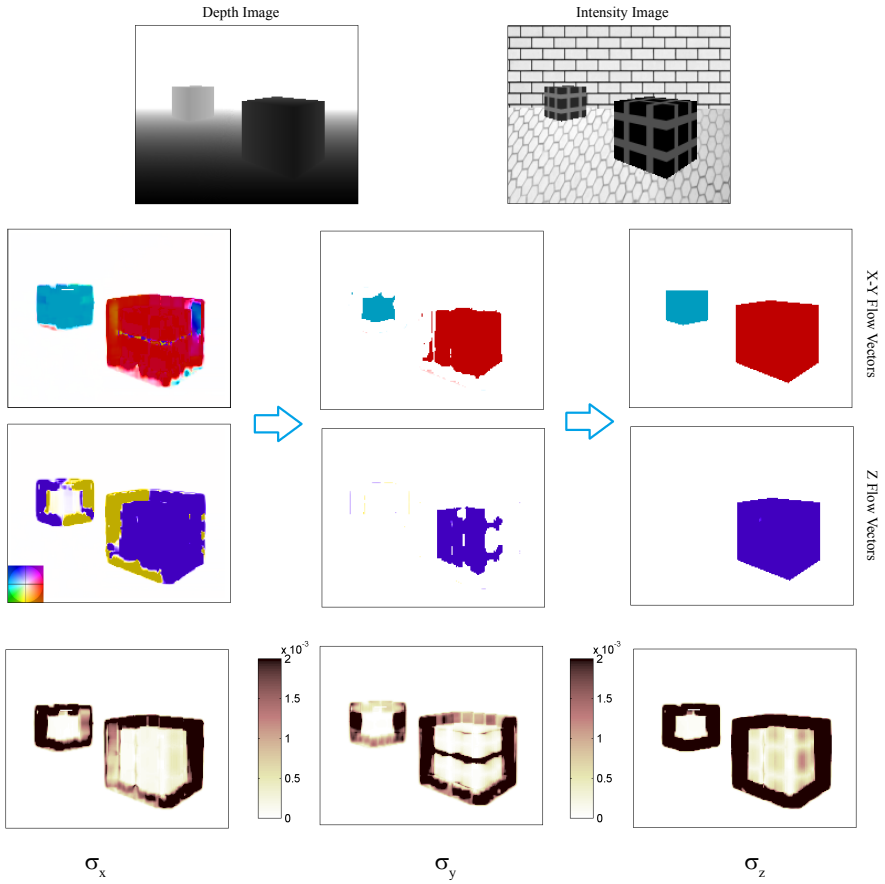

Figure 6.12: Illustration of the steps of motion estimation algorithm. *1st Row*: A frame from the *cubes* sequence. *2nd and 3rd Row*: $(\dot{X}, \dot{Y})$ and $\dot{Z}$ components of the motion vectors respectively. The left column shows the results of the local motion estimation step. In *4th Row* the standard deviation (in meters) of the three components of the estimated flow vector is given. Using a threshold on these values, less accurate flow vectors are filtered out as shown in the middle column. In the right column the regularized flow vectors are shown.

bottom row of Figure 6.12. Using a threshold, which is empirically chosen, these inaccurate flow vectors are filtered out. To achieve dense regularized flow vectors, the global regularization step is applied which gives the final 3D flow vectors. As the smoothing constraint was relaxed at intensity and depth discontinuity, the motion boundaries are sharp and consistent with the boundary of the moving cubes.

To further evaluate the algorithm's robustness to noise, we add Gaussian noise to the depth and intensity images of the synthetic scene. To add the noise, the depth and the intensity images are normalized to $[0, 1]$ using the maximum depth and maximum intensity, respectively. The resulting depth and intensity images have a common range of values. Subsequently, the flow

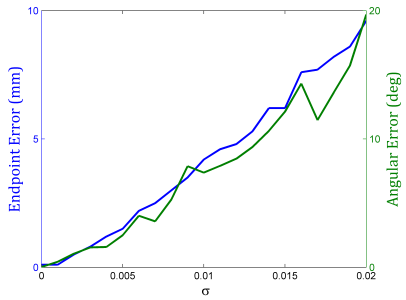

Figure 6.13: Endpoint Error (*blue*) and Angular Error (*green*) against the noise.

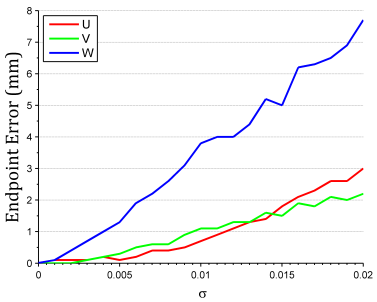

Figure 6.14: Endpoint Error for U, V and W of the flow vectors against the noise.

vectors are estimated and evaluated. Figure 6.13 shows the change in average EE and AE with increasing noise σ. Figure 6.14 shows the EE for each component of the flow vector separately, against the increasing noise values. In Figure 6.13 and 6.14, σ is the standard deviation of the additive Gaussian noise. As shown in Figure 6.14, the EE for $\dot{Z}$ is larger than the EE for $\dot{X}$ and $\dot{Y}$ for increasing σ, which is due to the fact that the motion along the viewing axis is harder to estimate, while looking at an image patch or local neighborhood.

Real World Scenes

The algorithm for estimating motion of independently moving objects is now evaluated on two real world scenes from ToF cameras: *trains* (Figure 6.15) and *people* (Figure 6.16). The frame rate of the videos is approximately 10 frames per second (fps) for *trains* and 5 fps for *people*. *Trains* consists of two toy trains moving on two rail tracks. The lateral extent of the scene is one meter. The depth ranges from 40 centimeters to one meter. The first train is on an elevated track and moves approximately diagonally from left to right in the image while the second train moves from top towards bottom of the image. *People* consists of three people, two of them walking approximately parallel to the image plane while a third person walks towards the camera in the later half of the video.

The GT motion for *trains* is computed by measuring the distance between two target pairs that are mounted on the rail tracks in the coordinate system attached to the camera. When assuming a constant linear velocity and rigid object motion, the GT flow vector of each train can be obtained by the time (i.e., number of frames) and the trains' traveling distances between the targets. The GT flow vectors for the train on the top and the train on the bottom are $[5.4, 4.4, -5.4]$ and $[0.6, 6.8, -7.2]$ (mm per frame), respectively. Due to complexity and inconsistency of the human motion, no GT is available for *people*.

In the local motion estimation step, the motion is only computed over pixels which change in time, by observing the change in depth and intensity in frames before and after the current frame. Furthermore, in the local motion step flow vectors on the background pixels of a depth discontinuities are removed as it is assumed that the background pixels will get occluded or disoccluded due to foreground motion. This is done by using a Laplacian of Gaussian filter which

Table 6.7: Quantitative evaluation of estimated motion vectors for calibrated (calib.) and uncalibrated (uncalib.) *trains*. Average Angular Error (AE) and Endpoint Error (EE).

trains	**calib**.	**uncalib**.
AE (deg.)	12.2	39
EE (mm)	3.05	6.9
EE in [U,V,W]	[1.4,1.3,2.4]	[2.1,2.2,6.2]

detect edges and indicates the the foreground and background pixels in the local neighboorhood of the depth image. Figure 6.15 shows the motion estimation result for five frames of the *trains* sequence along with the GT motion. The estimated flow vectors correspond well to the GT motion of the train. The boundaries of the trains appear sharp with little smoothing effects along some parts of the boundary.

To analyze the influence of calibration, the accuracy of the estimated motion is compared for the calibrated and uncalibrated *train* scene. The comparison of the estimated motion vectors with the GT motion vectors indicates a significantly higher EE in $\dot{Z}$ (Table 6.7) for uncalibrated data. The error values correspond to non stationary parts of the scene. Schmidt et al. [163] observed a similar behavior for $\dot{Z}$ and suspect systematic errors to be the reason for it. In fact, when investigating the results obtained from the calibrated [94] test scene, the EE of $\dot{Z}$ improves significantly. Moreover, Table 6.7 shows the overall AEs, the overall EEs and the EEs for the individual components of the estimated flow vectors for the uncalibrated and the calibrated test scene. When comparing the results of the calibrated train scene with the synthetic scene, the magnitudes of the errors are similar to errors resulting from higher levels of added noise. The applied calibration [94] does not completely remove the systematic effects that are causing distortions in distance observations. Furthermore, shadowing of the active illumination by foreground object and illumination fall off in intensity images violate the brightness constancy assumption and hence might introduce errors.

Figure 6.16 shows the motion estimation results for *people*. Considering the complexity of the scene, the motion estimation results appear well in correspondence with the actual motion. The direction of movement is estimated correctly and persons are delineated correctly. However, it can also be seen that fast, non-rigid motion of small structures that are different from the motion at similar depths (e.g., legs with different motion of body) is especially difficult to estimate. In this context, the low resolution and the high noise levels of ToF videos are challenging for motion estimation. Especially, the motion in depth $\dot{Z}$ is ill-determined. However, the results show that the regularization scheme performs well in determining dense and smooth flow fields over major parts of the independently moving objects. The advantage of simultaneous utilization of intensity and depth information is also evident in Figure 6.16, as in some areas of the scene, intensity gradients are higher while in some areas the depth gradients provide more information in estimation of the 3D motion vectors.

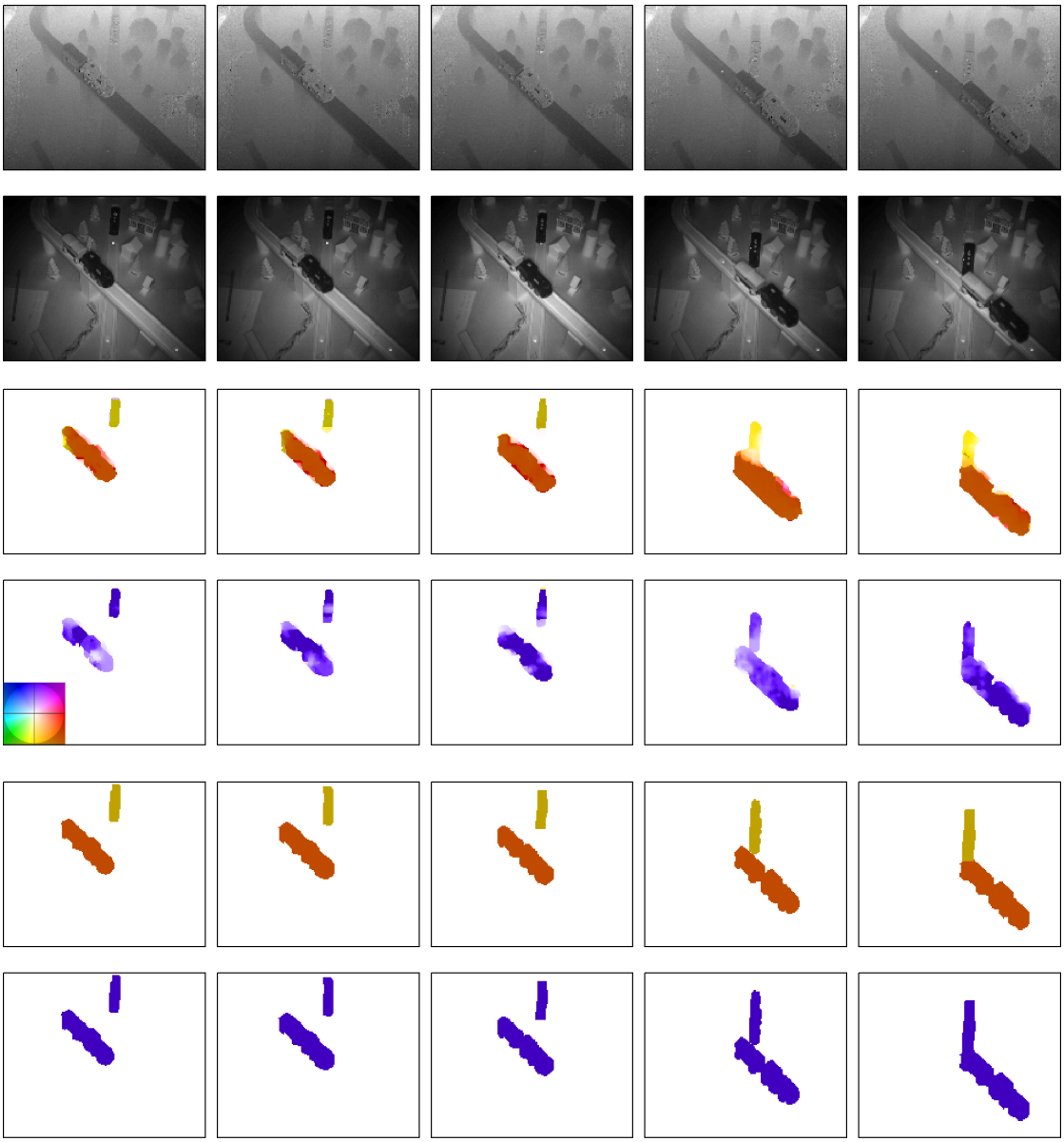

Figure 6.15: *1st Row*: Depth images from five frames of the *Train* sequence. *2nd Row* Corresponding intensity images. *3rd Row*: $(\dot{X}, \dot{Y})$ components of estimated motion vectors. *4th Row*: $\dot{Z}$ component of the estimated motion vectors. *5th Row*: $(\dot{X}, \dot{Y})$ components of GT motion vectors. *6th Row*: $\dot{Z}$ component of the GT motion vectors. GT motion vectors are generated from manual segmentation and using the GT motion. The motion vectors are color coded (color wheel) as: Hue encodes orientation and saturation encodes magnitude. $\dot{Z}$ is represented by only vertical axis of the color wheel.

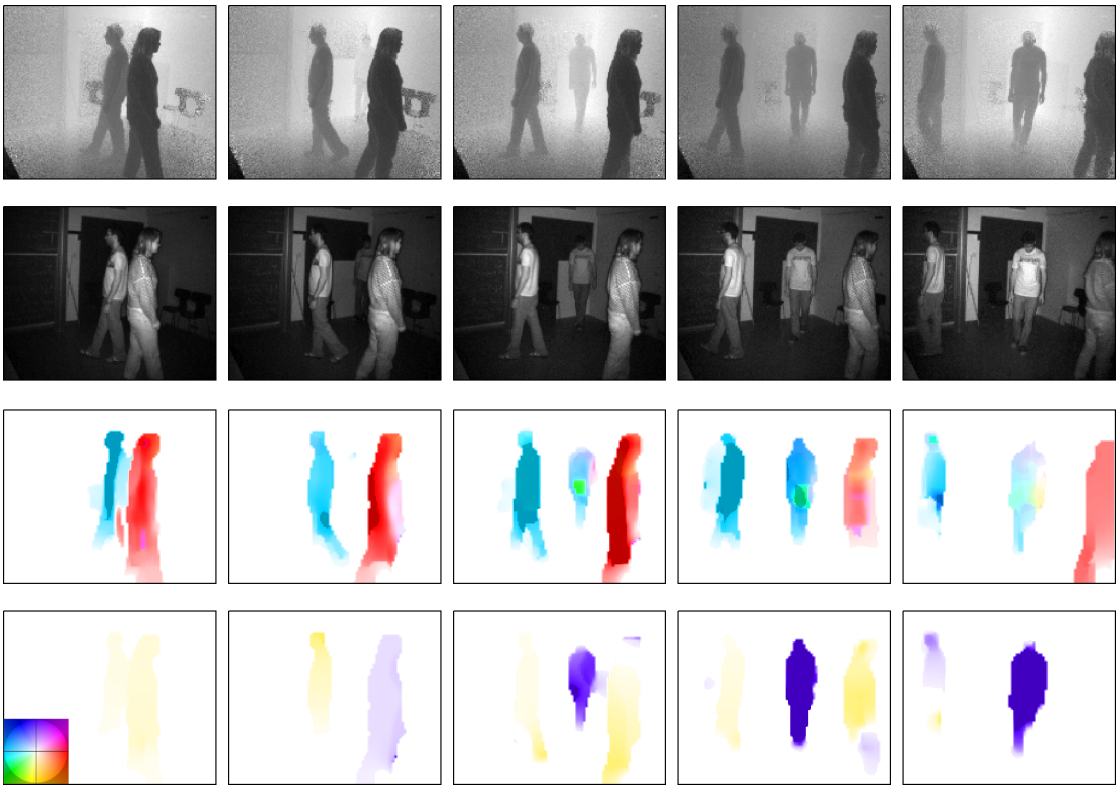

Figure 6.16: *1st Row*: Depth images for five frames of the *People* sequence. *2nd Row* Corresponding intensity images. *3rd Row*: $(\dot{X}, \dot{Y})$ components of estimated motion vectors. *4th Row*: $\dot{Z}$ component of the estimated motion vectors. The motion vectors are color coded (color wheel) as: Hue encodes orientation and saturation encodes magnitude. $\dot{Z}$ is represented by only vertical axis of the color wheel.

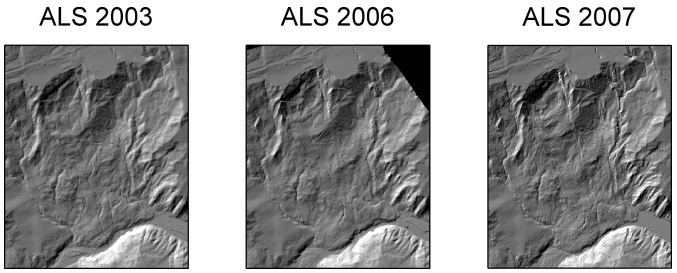

Figure 6.17: Shaded DTMs from ALS 2003–2007.

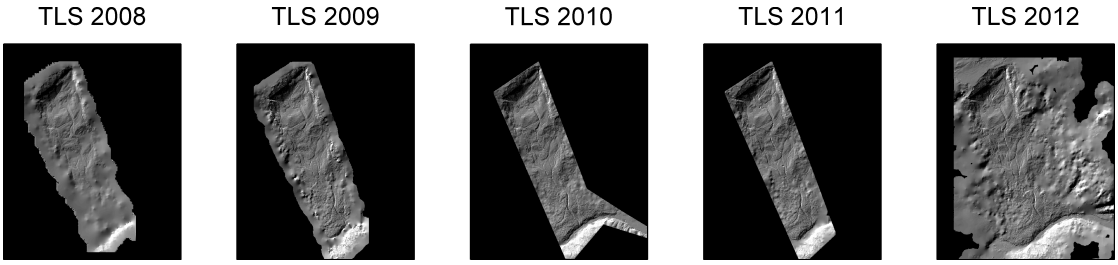

Figure 6.18: Shaded DTMs from TLS 2008–2012.

6.3 Motion Estimation of a Landslide

Motion as a result of natural phenomena like landslides and glacier movements is a very important topic due to its impact on environment and human life. Studying the changes in the surfaces requires acquisition of multi-temporal data of the subject area. The dynamics of the process defines the temporal resolution of the data acquisition. For studying the motion of slow moving landslides or glaciers, data can be acquired with the time difference of several months. The surface modeling for analysis of these natural processes is often based on remotely sensed data. Typical remote sensing techniques are based on photographs/images [157], airborne and terrestrial laser scanning [52, 157] and radar [188].

The subject study case is a landslide in Doren, Vorarlberg , Austria. This landslide has already been sketched in the historical maps of 19th century [52, 157]. In the recent years there were major movements in year 2005 and 2007 and since then the landslide is continuously evolving. Due to its proximity to human settlement, its a concern for the safety of local community. The landslide area has been measured on several epochs. Airborne LiDAR acquisitions were carried out from 2003 to 2007 while and TLS campaigns were carried out annually on the landslide site from 2008 to 2013.

LiDAR has often been used for investigation and characterization of landslides. The temporal data acquired from LiDAR sensor is used for detection and estimation of motion, susceptibility mapping and monitoring of landslides [87, 132, 185]. In current application, the interest is to estimate motion over the landslide, which is realized by applying range flow over the multi-

Figure 6.19: Orthophotos of the landslide area from year 2006 (left), 2007 (middle) and 2009 (right). Especially in the southern part of the landslide, efforts for an artificial drainage can be clearly recognised. Images by courtesy of Landesamt für Vermessung und Geoinformation Vorarlberg [34]

temporal surface models generated from LiDAR data. These surface models are generated by filtering out laser echoes from vegetation and artificial objects and computing the height at each grid point by fitting a plane to the points in the neighborhood [109]. The resulting height values over a regularly sampled grid or a raster gives the digital surface models of the landslide area. More details on the generation of the digital surface models from the laser scanning data can be found in [109, 156]

Range flow constraint is used to estimate motion on the landslide surface, which provides a quantitative measure of the movements in the landslide area .The available ALS and TLS data sets of the area don't provide a regular temporal sampling of the surface. Therefore, motion is estimated between each consecutive data sets only and no displacements rate over time are given because the landslide motion is inhomogeneous both in space and time. The motion patterns over a landslide are quite complex and result in varying motion on different parts of the landslide surface. Therefore, the global and temporal motion constraints are not suitable in such a scenario. The method for estimating motion is based on Lucas/Kanade type optical flow algorithms (Eq. (5.3)). A window of size 11×11 meters is chosen and a over determined systems of equations is solved using robust least squares estimator. In some areas of the landslides, motion is more than 10 m, therefore its necessary to use a coarse to fine warping scheme for motion estimation. Robust estimation is necessary due to presence of non uniform motion patterns e.g. at the scarp of the landslide there are large movements while the surface at the border remains quite stable.

To this end, it is essential to point out that the motion vectors computed using range flow do not necessarily represent motion of the mass or material. These motion vectors reflect surface changes which are results of geomorphic processes like debris flow, erosion, incision and

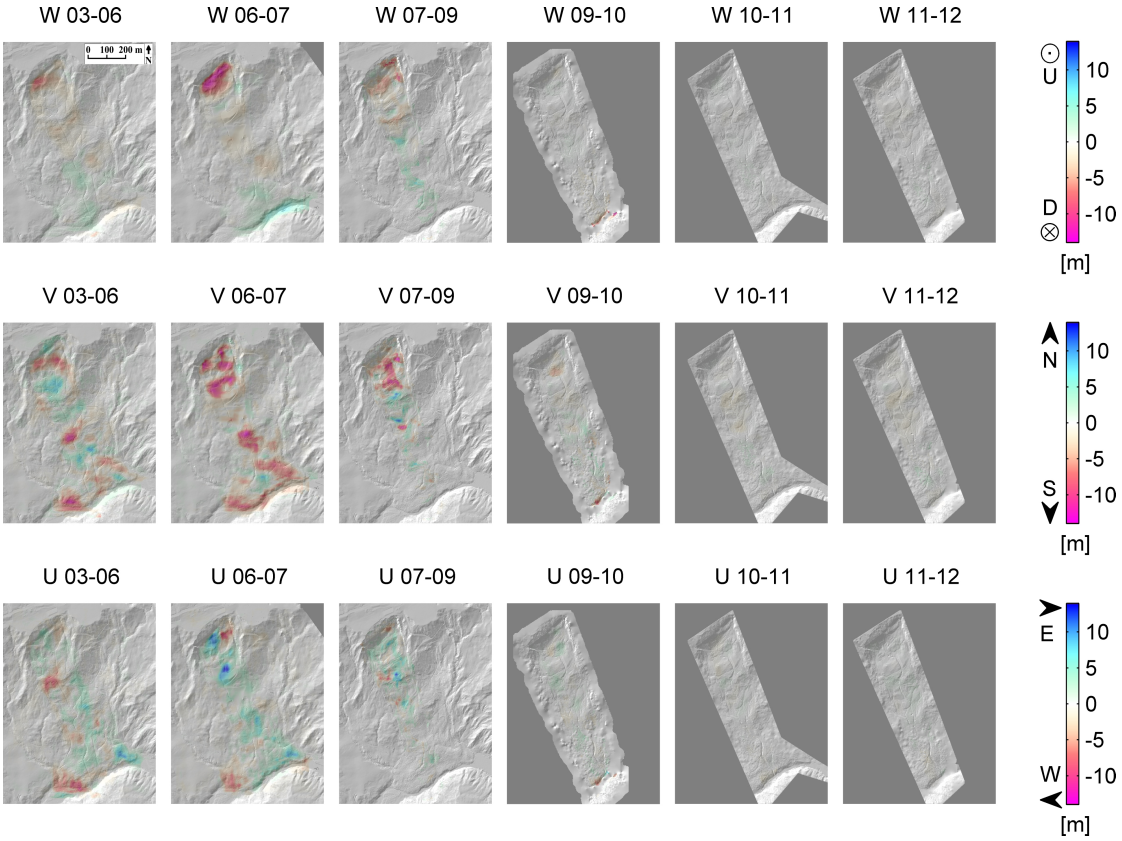

Figure 6.20: Motion vectors from 2003 to 2012. W ... elevation change, U ... east-west motion, V ... north-south motion. Numbers in the form NN-MM, indicate change from year 20NN to year 20MM. U = up (towards zenith), D = down (towards nadir).

the changes due to anthropogenic influence especially the construction of water drainage in landslide area. Furthermore, constraints like conservation of material have not been taken into consideration during this analysis. Loss of material may appear as a downward movement in the motion vectors and the accumulation of the material may result in apparent upward movement of the surface. This type of motion is apparent in the scarp area of the landslide where loss of material appears as downward motion.

Figure 6.20 shows the motion of the landslide from year 2003 to 2012. In the current context motion $(\dot{X}, \dot{Y}, \dot{Z}) = (\dot{U}, \dot{V}, \dot{W})$ corresponds to motion along west-east, south-north and the zenith direction. It can be seen that there were large movements (> 10 meters) during years 2003-2007, while since then the magnitude of the motion is relatively small, however the landslide still continues to evolve. Figure 6.21, gives a more detail look on the motion over the scarp of the landslide, where large movements occurred especially between the laser scanning campaigns of 2006 and 2007. On a qualitative basis, the motion vectors were manually compared by analyzing changes in the multi-temporal surface models and they appear well in

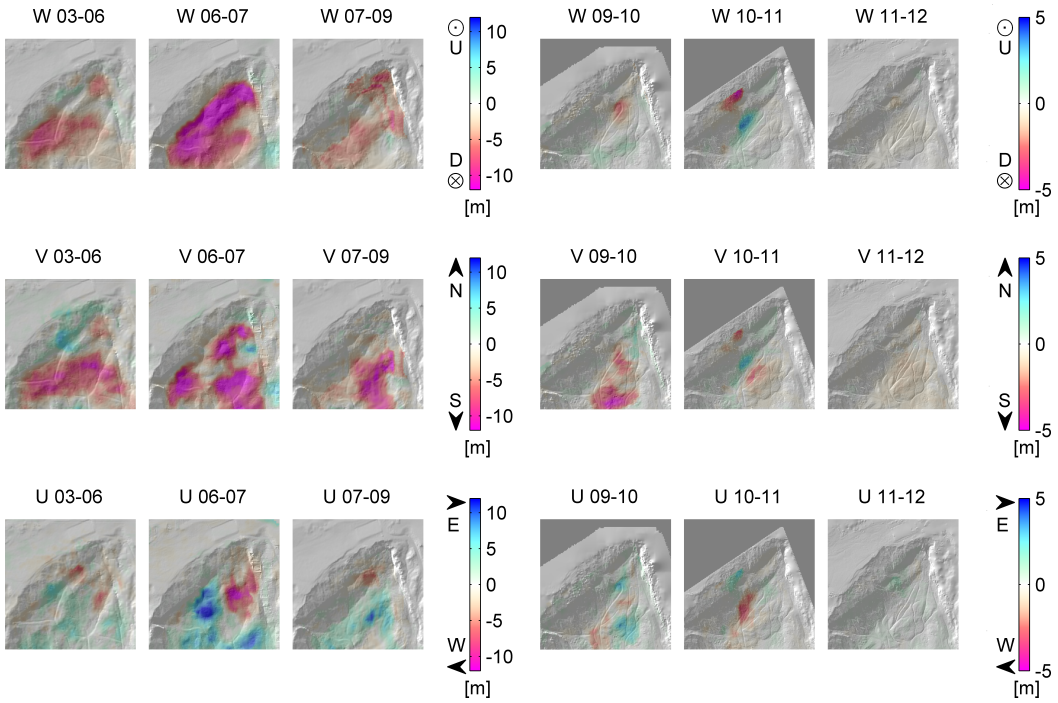

Figure 6.21: Motion vectors from 2003 to 2012 overlaid over the respective digital terrain models (DTMs) in the area of the scarp of the landslide

accordance with the visual analysis of the changes of the surface. However, in areas with high anthropogenic influence i.e. the area where the development of an artificial drainage system took place (see Figure 6.19), the motion vectors are not reliable as the surface is heavily altered due to human influence. Since March 2010, geodetic measurement campaigns have been undertaken by the local surveying authority (Landesamt für Vermessung und Geoinformation) to measure the locations of points (prisms) place on selected areas of the landslide [131]. Figure 6.22 shows the comparison of the estimated motion with the geodetic measurements of the points for two time intervals. In general the motion patterns from the estimated from range flow and geodetic points corresponds well to each other and the mean difference between the two vectors is around few decimeters. The prisms, which have been used as the target point, have been mounted on the steel rods or in some cases on tree trunks as shown in Figure 6.23. This creates a lever arm between the point position and the land surface. Due to movements in the surface these rods and tree trunks incline (as visible in Figure 6.23), which leads to a different point/prisms motion as compared to motion of the surface. Furthermore, the changes in the surface caused by processes like erosion may not show up in the observed movements of the points. From the on field experience, it appears that this may be the reason for the differences that are observed in the single point based motion estimation and motion estimation from range flow which uses a local surface patch.

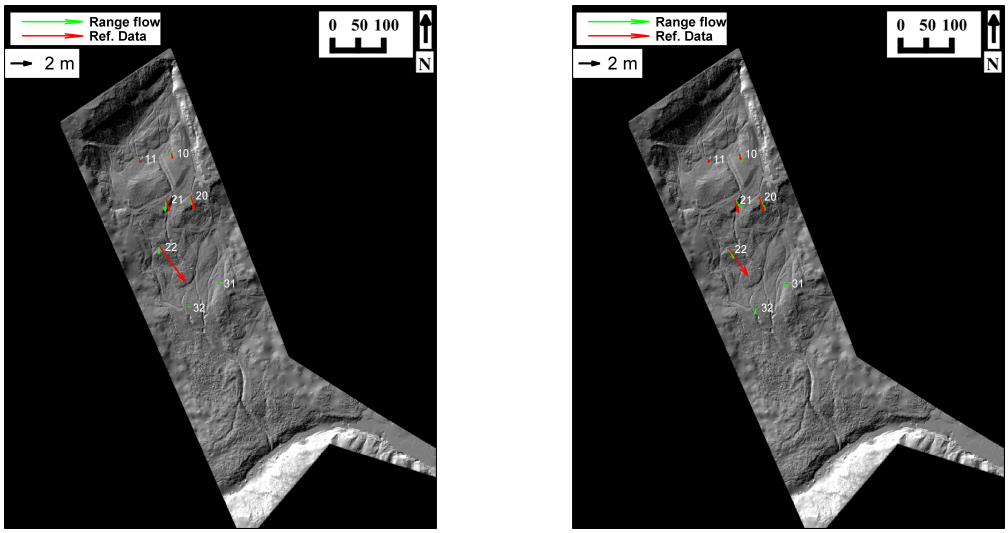

Figure 6.22: Motion vectors of the target points from reference data [131] and range flow overlaid over 2010 DTM

Figure 6.23: Reference target points of the Surveying Office of Vorarlberg, Austria [131]

Pts.	Range Flow 10–11 (m)	Ref. Vector 10–11 (m)	Difference 10–11 (m)	Range Flow 11–12 (m)	Ref. Vector 11–12 (m)	Difference 11–12 (m)
#11	0.15, −0.16, 0.46	0.05, −0.32, −0.16	0.10, 0.17, 0.62	0.05, −0.38, −0.07	0.06, −0.35, −0.15	−0.01, −0.03, 0.08
#22	−0.20, −0.58, −0.01	**2.41, −3.32, −0.21**	−2.61, 2.74, 0.20	0.41, −0.84, −0.25	1.76, −2.56, 0.05	**−1.34, 1.72, −0.31**
#21	−0.14, −1.15, −0.35	0.31, −1.10, −0.27	−0.44, −0.05, −0.08	0.55, −0.69, −0.46	0.35, −1.20, −0.21	0.21, 0.50, −0.25
#10	−0.15, −0.11, −0.10	0.10, −0.49, −0.02	−0.26, 0.38, −0.08	0.28, −0.59, −0.14	0.11, −0.56, 0.00	0.17, −0.02, −0.14
#32	0.07, 0.07, 0.21	0.01, −0.03, −0.00	0.06, 0.10, 0.22	−0.01, −0.29, −0.13	0.01, −0.03, 0.00	−0.02, −0.26, −0.13
#20	0.22, −0.86, −0.32	0.38, −1.31, −0.32	−0.16, 0.46, 0.01	0.64, −1.26, −0.46	0.48, −1.41, −0.33	0.16, 0.15, −0.13
#31	0.23, 0.23, 0.14	−0.00, −0.03, −0.00	0.24, 0.26, 0.14	−0.08, −0.36, −0.07	−0.01, −0.03, 0.00	−0.07, −0.33, −0.08

Table 6.8: Comparison between the estimated motion (U,V,W) and the reference data.

CHAPTER 7

Conclusions

This thesis presented new methods to estimate motion from the integration of range and intensity data using image sequences from different types of 3D sensors. Common to these sensors is the simultaneous, synchronous frame wise acquisition of gray-scale or color information and depth information at video frame rates (typically 30 Hz). These sensors are available only since a few years and neither their resolution nor their accuracy can be compared to professional, e.g. aerial photogrammetric, cameras or laser scanners.

Motion of a camera and motion of independently moving objects are both discussed in the thesis. At the core of the proposed methods is the optical flow and range flow constraints, which defines the transformation of the pixels from one frame to another. Due to similarity of the two constraints, they can be well integrated to solve an adjustment problem which simultaneously utilizes the range and intensity information as available in the state of the art 3D sensors. The range and optical flow constraints can be written for each pixel, thus dense pixel wise information have been exploited.

In Chapter 4, first a method for relative orientation of images from a moving camera based on integration of optical flow and range flow constraints was presented. Using dense pixel wise information, image features like corners, edges and geometrical features like intersecting planes are all implicitly taken into consideration, while estimating motion. Therefore, the algorithm is able to automatically and robustly estimate motion of the camera in scenes with varying amount of texture and varying geometry in presence of motion blur and illumination changes. This was shown in the evaluation results on the sequences from RGB-D SLAM dataset and sequences from ToF cameras. It was also shown, that if an independent moving object is present in the data and the dominant motion is due to moving camera, the pixels belonging to the independent moving object are correctly detected as outliers in the robust adjustment. Furthermore, the coarse to fine strategy was applied for estimating relatively large motions. In the sequences from RGB-D dataset there are rotations upto 50 deg/sec, which results in image motion of more than 20 pixels in between consecutive frames. The results showed that these large motion were accurately estimated using the coarse to fine strategy.

In order to compute the trajectory of the camera the relative orientations are transformed to a common coordinate system. But as the relative transformations do not utilize global information the concatenation of relative orientations accumulates error due to the drift. To obtain a globally consistent trajectory a method utilizing relative orientation constraints obtained from range flow and optical flow in bundle adjustment is proposed. Bundle adjustment is performed using three groups of observations: Point correspondences, depth observations of these corresponding points and estimated relative orientations between consecutive frames. The SURF features are used for finding the corresponding or tie points between images. The feature matching is performed using the *keyframe* strategy, which helps to identify the same features points over a number of images and also helps to detect loop closures. The unknowns in the bundle adjustment are the camera orientations and the 3D positions of the feature points. As the bundle adjustment is a non linear optimization, approximate values of the unknown are required for initialization of bundle adjustment. The analysis showed that the relative orientations concatenated into a common coordinate system provides a good initial estimates of the camera orientation.

The weighting of the three observations groups in the adjustment is an important task. It is shown that the covariance estimates from the least squares solution of relative orientation are orders of magnitude too optimistic. Using variance component analysis in bundle adjustment, better estimates of the accuracy of the observations can be obtained. The results of variance component estimation showed that the original accuracy estimates of the camera relative orientation were highly optimistic and the revised accuracy estimates show the standard deviation of camera translation parameters to be in mm range. It was also observed that the accuracy model used for depth observations of the Kinect sensor also gives an optimistic accuracy estimate for the evaluation sequences used in this thesis. The weighting of these individual observations is essential for obtaining accurate estimates of the unknown motion and its accuracy estimates.

The quantitative results in the form of relative pose error and absolute trajectory error on the RGB-D SLAM dataset showed that the accuracy of the relative orientation method and bundle adjustment is comparable to the accuracy of the state of the art SLAM algorithm. In fact for several sequences the proposed method achieves better accuracy than the RGB-D SLAM algorithm. The proposed method provides estimates of the accuracy of the trajectory, which using the law of error propagation can be brought forward to the points observed in object space, which can be considered in modeling the scene.

In Chapter 5 3D motion estimation of independently moving objects using range and optical flow was presented. Similar, to the method of relative orientation, range flow and optical flow constraints for each pixel is used to estimate 3D motion. This method consisted of two steps, in the first step only local information was used to estimate flow at each pixel, while in the second step a global regularization was performed to obtain smooth dense motion vectors. One advantage of the two steps approach is that it leads to a linear system of equations and the outliers are removed by performing robust adjustment. The results showed that using this strategy, dense 3D flow vectors with sharp motion boundaries are achievable.

For future work, the weighting of the relative orientation terms in the bundle adjustment can be further investigated. In this work, the translational and rotational parameters of relative orientation were placed together in one group of observation equations. It can be investigated, how the variance factors for translation and rotational groups behave if they are placed in separate

groups. To achieve robustness against illumination changes, some optical flow algorithms use gradient constancy assumption in addition to brightness constancy assumption. Therefore, the use of gradient constancy assumption in relative orientation method can be investigated as well. As the range cameras continue to develop, the resolution and the accuracy of these cameras is expected to increase in the future. Due to increase in resolution, its essential to take into account the computational aspects of the algorithm. Therefore, in the global regularization method and in bundle adjustment and subsequently variance component analysis, special handling of large amount of data needs to be implemented.

Overall, the thesis demonstrated the advantages of proper stochastic modeling of the observations and feasibility (and advantages) of simultaneous consideration of intensity and range measurements.

Bibliography

[1] Y. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates in close-range photogrammetry. In *ASP Symposium on Close-Range Photogrammetry*, 1971.

[2] F. Ackermann. Digital image correlation: performance and potential application in photogrammetry. *The Photogrammetric Record*, 11(64):429–439, 1984.

[3] G. Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, 1989.

[4] P. Agarwal, W. Burgard, and C. Stachniss. Helmert's and bowie's geodetic mapping methods and their relation to graph-based slam. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

[5] D. Akca. *Least squares 3D surface matching*. PhD thesis, Eidgenössische Technische Hochschule ETH Zürich, No. 17136, 2007.

[6] A. Amiri-Simkooei and S. Jazaeri. Weighted total least squares formulated by standard least squares theory. *Journal of Geodetic Science*, 2(2):113–124, 2012.

[7] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[8] P. Baker, C. Fermuller, Y. Aloimonos, and R. Pless. A spherical eye from multiple cameras (makes better models of the world). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[9] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[10] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[11] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[12] P. Besl and N. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[13] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proceedings of the International Conference on Computer Vision*, pages 231–236, 1993.

[14] G. Blaha. Inner adjustment constraints with emphasis on range observations, depart. Technical Report No. 148, Department Of Geodetic Science, Ohio State University, 1971.

[15] M. Bleyer and M. Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing: Image Communication*, 22(2):127–143, 2007.

[16] N. Brosch, A. Hosni, C. Rhemann, and M. Gelautz. Spatio-temporally coherent interactive video object segmentation via efficient filtering. In *Proceedings of the Joint 34th DAGM and 36th OAGM Symposium*, pages 418–427, 2012.

[17] N. Brosch, C. Rhemann, and M. Gelautz. Segmentation-based depth propagation in videos. In *Proceedings of the ÖAGM/AAPR Workshop*, volume 2011, pages 1–8, 2011.

[18] D. C. Brown. The bundle adjustment—progress and prospects. *International Archives Photogrammetry*, 21(3):1–1, 1976.

[19] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision*, pages 25–36, 2004.

[20] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[21] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr. Variational optical flow computation in real time. *IEEE Transactions on Image Processing*, 14(5):608–615, 2005.

[22] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[23] B. Büttgen and P. Seitz. Robust optical Time-of-Flight range imaging based on smart pixel structures. *IEEE Transactions on Circuits and Systems*, 55(6):1512–1525, 2008.

[24] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 3, pages 2724–2729, 1991.

[25] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu. Large displacement optical flow from nearest neighbor fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2450, 2013.

[26] J. Chow and D. Lichti. Photogrammetric bundle adjustment with self-calibration of the primesense 3D camera technology: Microsoft kinect. *IEEE Access*, 1:465–474, 2013.

[27] F. Coleca, T. Martinetz, and E. Barth. Gesture interfaces with depth sensors. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 207–227. Springer, 2013.

[28] M. Cramer, D. Stallmann, and N. Haala. Direct georeferencing using gps/inertial exterior orientations for photogrammetric applications. *International Archives of Photogrammetry and Remote Sensing*, 33(B3/1; PART 3):198–205, 2000.

[29] K. Dannilidis and H.-H. Nagel. The coupling of rotation and translation in motion estimation of planar surfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 188–193. IEEE, 1993.

[30] M. Davidovic, J. Seiter, M. Hofbauer, W. Gaberl, and H. Zimmermann. A background light resistant tof range finder with integrated pin photodiode in 0.35um cmos. In *Proceedings SPIE 8791, Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*, pages 87910R–87910R–6, 2013.

[31] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1403–1410. IEEE, 2003.

[32] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[33] B. Drayton, D. Carnegie, and A. Dorrington. Variable frame time imaging for indirect time of flight range imaging cameras. In *Proceedings of the IEEE International Conference on Instrumentation and Measurement Technology*, pages 609–613, 2013.

[34] P. Drexel and M. Seebacher. Einmal ist keinmal – die Anwendung von Luftbild-/Laserscanning-/Geodatenzeitreihen in der Vorarlberger Landesverwaltung. In K. Hanke and T. Weinold, editors, *17. Internationale Geodätische Woche Obergurgl*, pages 50–55, Berlin, Offenbach, 2013. Wichmann Verlag.

[35] D. Droeschel, S. May, D. Holz, P. Ploeger, and S. Behnke. Robust ego-motion estimation with tof cameras. In *European Conference on Mobile Robots*, 2009.

[36] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, 1997.

[37] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1691–1696, 2012.

[38] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1):177–187, Feb 2014.

[39] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3D visual SLAM with a hand-held RGB-D camera. In *Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, 2011.

[40] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. *Photogrammetric Computer Vision*, 2, 2006.

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

[42] W. Förstner. On the geometric precision of digital correlation. *International Archives Photogrammetry & Remote Sensing*, 24(3):176–189, 1982.

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

[44] D. Fritsch. Photogrammetry as a tool for detecting recent crustal movements. *Tectonophysics*, 130(1):407–420, 1986.

[45] S. Fuchs. Multipath interference compensation in Time-of-Flight camera images. In *Proceedings of the International Conference on Pattern Recognition*, pages 3583–3586, 2010.

[46] G. V. G. Sithole. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59:85–101, 2004.

[47] C. Garbe, H. Spies, and B. Jähne. Mixed OLS-TLS for the estimation of dynamic processes with a linear source term. In L. Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 463–471. Springer Berlin Heidelberg, 2002.

[48] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[49] S. Ghuffar, N. Brosch, N. Pfeifer, and M. Gelautz. Motion segmentation in videos from time of flight cameras. In *Proceedings of the International Conference on Systems, Signals and Image Processing*, pages 328–332, 2012.

[50] S. Ghuffar, N. Brosch, N. Pfeifer, and M. Gelautz. Motion estimation and segmentation in depth and intensity videos. *Integrated Computer-Aided Engineering*, 21(3):203–218, 2014.

[51] S. Ghuffar, C. Ressl, and N. Pfeifer. Relative orientation of videos from range imaging cameras. In *Proceedings of the SPIE 8791, Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*, page 879114, 2013.

[52] S. Ghuffar, B. Székely, A. Roncat, and N. Pfeifer. Landslide displacement monitoring using 3D range flow on airborne and terrestrial lidar data. *Remote Sensing*, 5(6):2720–2745, 2013.

[53] G. H. Golub and R. J. Plemmons. Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. *Linear Algebra and Its Applications*, 34:3–28, 1980.

[54] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[55] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, and P. Arias. Metrological evaluation of microsoft kinect and asus xtion sensors. *Measurement*, 46(6):1800 – 1806, 2013.

[56] J. Gottfried, J. Fehr, and C. Garbe. Computing range flow from multi-modal Kinect data. In *Advances in Visual Computing*, volume 6938 of *Lecture Notes in Computer Science*, pages 758–767. Springer, 2011.

[57] S. Granshaw. Bundle adjustment methods in engineering photogrammetry. *The Photogrammetric Record*, 10(56):181–207, 1980.

[58] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 273–278, 2010.

[59] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems*, 2007.

[60] A. Gruen. Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.

[61] A. Gruen. Development and status of image matching in photogrammetry. *The Photogrammetric Record*, 27(137):36–57, 2012.

[62] A. W. Gruen and E. P. Baltsavias. Geometrically constrained multiphoto matching. *Photogrammetric Engineering and Remote Sensing*, 54(5):633–641, 1988.

[63] A. Grün and D. Akca. Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):151–174, 2005.

[64] M. Gsandtner and H. Kager. An out-of-core solution of normal equations providing also accuracy and reliability data. In *Proceedings of the XVI th ISPRS Congress*, volume 27, pages 52–59, 1988.

[65] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[66] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2001.

[67] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill. 3D pose tracking with linear depth and brightness constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 206–213, 1999.

[68] C. Heipke, K. Jacobsen, and H. Wegmann. Analysis of the results of the OEEPE test 'integrated sensor orientation'. *OEEPE Official Publication*, 43:31–49, 2002.

[69] C. Heipke, K. Jacobsen, H. Wegmann, Ø. Andersen, and B. Nilsen. Integrated sensor orientation-an OEEPE test. *International Archives of Photogrammetry and Remote Sensing*, 33(B3/1; PART 3):373–380, 2000.

[70] F. R. Helmert. *Die Ausgleichungsrechnung nach der Methode der kleinsten Quadrate: mit Anwendungen auf die Geodäsie und die Theorie der Messinstrumente*. BG Teubner, 1872.

[71] F. R. Helmert. Die mathematischen und physikalischen Theorieen der höheren Geodäsie. *Leipzig, BG Teubner, 1880-94.*, 1, 1880.

[72] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[73] A. Heyden. A common framework for multiple view tensors. In *Proceedings of the European Conference on Computer Vision-Volume I*, pages 3–19, 1998.

[74] M. J. Hinich and P. P. Talwar. A simple method for robust regression. *Journal of the American Statistical Association*, 70(349):113–119, 1975.

[75] B. Horn. *Robot Vision*. MIT electrical engineering and computer science series. MIT Press, 1986.

[76] B. Horn and J. Harris. Rigid body motion from range image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, 53(1):1–13, 1991.

[77] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981.

[78] B. Horn and E. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, 1988.

[79] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Opical Society of America A*, 4:629–642, 1987.

[80] B. K. Horn. Determining optical flow: a retrospective. *Artificial Intelligence*, 59:81–87, 1993.

[81] B. K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984.

[82] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of Amercia A*, 5(7):1127–1135, 1988.

[83] T. Huang, S. Blostein, and E. Margerum. Least-squares estimation of motion parameters from 3-D point correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 10, pages 112–115, 1986.

[84] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[85] M. Irani and P. Anandan. About direct methods. *Vision Algorithms: Theory and Practice*, pages 267–277, 2000.

[86] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 559–568, 2011.

[87] M. Jaboyedoff, T. Oppikofer, A. Abellán, M. Derron, A. Loye, R. Metzger, and A. Pedrazzini. Use of LIDAR in landslide investigations: A review. *Natural Hazards*, 61(1):5–28, 2012.

[88] A. E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9-10):635–651, 1998.

[89] G. Jones. Accurate and computationally-inexpensive recovery of ego-motion using optical flow and range flow with extended temporal support. In *Proceedings of the British Machine Vision Conference*, 2013.

[90] G. Jones. Combining optical flow and range flow to recover RGBD sensor ego-motion. In *RGB-D: Advanced Reasoning with Depth Cameras, Berlin, Germany*, 2013.

[91] G. A. Jones and G. Hunter. Spatio-temporal support for range flow based ego-motion estimators. In *Computer Analysis of Images and Patterns*, pages 531–538. Springer, 2013.

[92] B. Jutzi. Investigations on ambiguity unwrapping of range images. *International Archives of Photogrammetry and Remote Sensing*, 38,Part 3/W8:265–270, 2009.

[93] H. Kager, F. Rottensteiner, M. Kerschner, and P. Stadler. *ORPHEUS 3.2.1 User Manual*. Institute of Photogrammetry and Remote Sensing, Vienna University of Technology, Austria, 2002.

[94] W. Karel. Integrated range camera calibration using image sequences from hand-held operation. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 37 (Part B5)*, pages 945–952, Beijing, China, 2008. ISPRS.

[95] W. Karel, P. Dorninger, and N. Pfeifer. In situ determination of range camera quality parameters by segmentation. In *Proceedings of the International Conference on Optical 3-D Measurement Techniques*, pages 109–116, 2007.

[96] W. Karel, S. Ghuffar, and N. Pfeifer. Quantifying the distortion of distance observations caused by scattering in Time-of-Flight range cameras. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5), 2010.

[97] W. Karel, S. Ghuffar, and N. Pfeifer. Modelling and compensating internal light scattering in time of flight range cameras. *The Photogrammetric Record*, 27(138):155–174, 2012.

[98] R. Kaufmann, M. Lehmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, T. Oggier, N. Blanc, P. Seitz, G. Gruener, et al. A Time-of-Flight line sensor: development and application. In *Proceedings SPIE, volume 5459*, pages 192–199. International Society for Optics and Photonics, 2004.

[99] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[100] J.-S. Kim, M. Hwangbo, and T. Kanade. Motion estimation using multiple non-overlapping cameras for small unmanned aerial vehicles. In *IEEE International Conference on Robotics and Automation*, pages 3076–3081. IEEE, 2008.

[101] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[102] K.-R. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer, 1999.

[103] G. B. Kolata. Geodesy: dealing with an enormous computer task. *Science*, 200(4340):421–466, 1978.

[104] K. Konolige. Large-scale map-making. In *Proceedings of the National Conference on Artificial Intelligence*, pages 457–463, 2004.

[105] K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[106] K. Konolige and P. Mihelich. Technical description of kinect calibration, 2011. accessed on 4th Feb. 2014.

[107] K. Kraus. *Photogrammetrie – Verfeinerte Methoden und Anwendungen*, volume 2. Dümmler-Verlag, Bonn, 3 edition, 1996.

[108] K. Kraus. *Photogrammetry – Geometry from Images and Laser Scans*. De Gruyter, 2 edition, 2007.

[109] K. Kraus and N. Pfeifer. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53:193–203, 1998.

[110] K. Kraus, C. Ressl, and A. Roncat. Least squares matching for airborne laser scanner data. In *5th International Symposium Turkish-German Joint Geodetic Days, Berlin, March*, pages 29–31, 2006.

[111] R. Lange. *3D Time-of-Flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*. PhD thesis, University of Siegen, 2000.

[112] R. Lange, P. Seitz, A. Biber, and S. C. Lauxtermann. Demodulation pixels in ccd and cmos technologies for Time-of-Flight ranging. In *Electronic Imaging*, pages 177–188. International Society for Optics and Photonics, 2000.

[113] D. D. Lichti, C. Kim, and S. Jamtsho. An integrated bundle adjustment approach to range camera geometric self-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(4):360 – 368, 2010.

[114] D. D. Lichti, X. Qi, and T. Ahmed. Range camera self-calibration with scattering compensation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 74(0):101 – 109, 2012.

[115] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[116] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[117] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[118] T. Luhmann, S. Robson, S. Kyle, and I. Harley. *Close Range Photogrammetry: Principles, Methods and Applications*. Whittles, 2006.

[119] Y. Ma, S. Soatto, J. Koseck, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2010.

[120] A. A. Markov. *Wahrscheinlichkeitsrechnung*. BG Teubner, 1912.

[121] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 211(1183):151–180, 1981.

[122] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[123] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[124] S. May, D. Droeschel, D. Holz, S. Fuchs, E. Malis, A. Nüchter, and J. Hertzberg. Three-dimensional mapping with Time-of-Flight cameras. *Journal of Field Robotics*, 26(11-12):934–965, 2009.

[125] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126 – 143, 2001.

[126] J. McGlone, E. Mikhail, J. Bethel, and Mullen. *Manual of Photogrammetry*. American Society for Photogrammetry and Remote Sensing, fifth edition, 2004.

[127] MESA Imaging AG. http://www.mesa-imaging.ch/. Accessed: 2014-02-27.

[128] E. M. Mikhail, J. S. Bethel, and J. C. McGlone. *Introduction to Modern Photogrammetry*, volume 1. John Wiley & Sons Inc, 2001.

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

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

[131] M. Mittelberger. Vom Nutzen bewegter Festpunkte und verrückter Grenzen. In A. Grimm-Pitzinger and T. Weinold, editors, *16. Internationale Geodätische Woche Obergurgl*, pages 85–89, Berlin, Offenbach, 2011. Wichmann Verlag.

[132] O. Monserrat and M. Crosetto. Deformation measurement using terrestrial laser scanning data and least squares 3D surface matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):142 – 154, 2008.

[133] F. Mufti and R. Mahony. Statistical analysis of signal measurement in Time-of-Flight cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(5):720–731, 2011.

[134] H.-H. Nagel. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *International Journal of Computer Vision*, 15(3):271–288, 1995.

[135] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):565–593, Sept 1986.

[136] F. Neitzel and S. Petrovic. Total least squares (TLS) im kontext der ausgleichung nach kleinsten quadraten am beispiel der ausgleichenden geraden. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, 133:141–148, 2008.

[137] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE international Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[138] L. Ng and V. Solo. Errors-in-variables modeling in optical flow estimation. *IEEE Transactions on Image Processing*, 10(10):1528–1540, 2001.

[139] W. Niemeier. *Ausgleichsrechnung, Statistische Auswertemethoden*. de Gruyter Textbook, 2008.

[140] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[141] D. Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.

[142] J. Oliensis. A linear solution for multiframe structure from motion. In *Proceedings of the Image Understanding Workshop*, pages 1225–1231, 1994.

[143] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2262–2269, 2006.

[144] J. Otepka, S. Ghuffar, C. Waldhauser, R. Hochreiter, and N. Pfeifer. Georeferenced point clouds: A survey of features and point cloud management. *ISPRS International Journal of Geo-Information*, 2(4):1038–1065, 2013.

[145] M. Otte and H.-H. Nagel. Optical flow estimation: Advances and comparisons. In *Proceedings of the European Conference on Computer Vision*, pages 49–60, 1994.

[146] H. Papo and A. Perelmuter. Free net analysis in close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 48(4):571–576, 1982.

[147] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

[148] K. Pulli. Multiview registration for large data sets. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.

[149] J. Quiroga, F. Devernay, and J. Crowley. Scene flow by tracking in intensity and depth data. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 50–57, 2012.

[150] F. Remondino and D. Stoppa. *ToF Range-Imaging Cameras*. Springer, 2012.

[151] C. Ressl. *Geometry, Constraints and Computation of the Trifocal Tensor*. PhD thesis, Vienna University of Technology, 2003.

[152] C. Ressl, H. Kager, and G. Mandlburger. Quality Checking Of ALS Projects Using Statistics Of Strip Differences. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 37 (Part B3b)*, pages 253–260, 2008.

[153] C. Ressl, N. Pfeifer, and G. Mandlburger. Appyling 3D affine transformation and least squares matching for airborne laser scanning strips adjustment without GNSS/INS trajectory data. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5/W12)*, Calgary, Canada, 2011.

[154] A. Roncat, G. Bergauer, and N. Pfeifer. B-spline deconvolution for differential target cross-section determination in full-waveform laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4):418–428, 2011.

[155] A. Roncat, C. Briese, J. Jansa, and N. Pfeifer. Radiometrically calibrated features of full-waveform lidar point clouds based on statistical moments. *IEEE Geoscience and Remote Sensing Letters*, 11(2):549–553, Feb 2014.

[156] A. Roncat, P. Dorninger, G. Molnár, B. Székely, A. Zámolyi, T. Melzer, N. Pfeifer, and P. Drexel. Influences of the Acquisition Geometry of different Lidar Techniques in High-Resolution Outlining of microtopographic Landforms. In *Fachtagung Computerorientierte Geologie – COGeo 2010*, 2010.

[157] A. Roncat, S. Ghuffar, B. Székely, P. Dorninger, S. Rasztovits, M. Mittelberger, Z. Koma, D. Krawczyk, and N. Pfeifer. A natural laboratory - terrestrial laser scanning and auxiliary measurements for studying an active landslide. In *Proceedings*, 2013. invited; talk: 2nd Joint International Symposium on Deformation Monitoring (JISDM), Nottingham, UK; 2013-09-09 – 2013-09-10.

[158] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[159] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[160] B. Schaffrin and A. Wieser. On weighted total least-squares adjustment for linear regression. *Journal of Geodesy*, 82(7):415–421, 2008.

[161] B. Schaffrin and A. Wieser. Total least-squares adjustment of condition equations. *Studia Geophysica et Geodaetica*, 55(3):529–536, 2011.

[162] H. Scharr and H. Spies. Accurate optical flow in noisy image sequences using flow adapted anisotropic diffusion. *Signal Processing: Image Communication*, 20(6):537–553, 2005.

[163] M. Schmidt, M. Jehle, and B. Jahne. Range flow estimation based on photonic mixing device data. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):380–392, 2008.

[164] T. Schuchert, T. Aach, and H. Scharr. Range flow for varying illumination. In *Proceedings of the European Conference on Computer Vision: Part I*, pages 509–522, 2008.

[165] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems*, volume 2, page 4, 2009.

[166] J. Seiter, M. Hofbauer, M. Davidovic, S. Schidl, and H. Zimmermann. Correction of the temperature induced error of the illumination source in a Time-of-Flight distance measurement setup. In *IEEE Sensors Applications Symposium*, pages 84–87, 2013.

[167] G. C. Sharp, S. W. Lee, and D. K. Wehe. Multiview registration of 3D scenes by minimizing error between coordinate frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1037–1050, 2004.

[168] M. Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge Aerospace Series. Cambridge University Press, 1997.

[169] E. Simoncelli. Design of multi-dimensional derivative filters. In *Proceedings of the IEEE International Conference Image Processing*, volume 1, pages 790–794 vol.1, Nov 1994.

[170] A. Singh. An estimation-theoretic framework for image-flow computation. In *Proceedings, Third International Conference on Computer Vision.*, pages 168–177, 1990.

[171] J. Smisek, M. Jancosek, and T. Pajdla. 3D with kinect. In *Consumer Depth Cameras for Computer Vision*, pages 3–25. Springer, 2013.

[172] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[173] K. Snow. *Topics in Total Least-Squares Adjustment within the Errors-In-Variables Model: Singular Cofactor Matrices and Prior Information*. PhD thesis, Geodetic Science Program, School of Earth Sciences, The Ohio State University, Columbus, 2012.

[174] H. Spies and J. Barron. Evaluating the range flow motion constraint. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 517–520, 2002.

[175] H. Spies, H. Haußecker, B. Jähne, and J. Barron. Differential range flow estimation. In *Proceedings of the Annual Symposium of the German Association for Pattern Recognition (DAGM)*, pages 309–316, 1999.

[176] H. Spies, B. Jahne, and J. Barron. Regularized range flow. In *Proceedings of the European Conference on Computer Vision: Part II*, pages 785–799, 2000.

[177] H. Spies, B. Jahne, and J. Barron. Range flow estimation. *Computer Vision and Image Understanding*, 85:209–231, 2002.

[178] H. Spies, B. Jahne, and J. L. Barron. Dense range flow from depth and intensity data. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 131–134, 2000.

[179] G. Strang. *Introduction to Linear Algebra*. SIAM, 2003.

[180] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the International Conference on Intelligent Robot Systems*, Oct. 2012.

[181] P. Sturm, W. Triggs, et al. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the European Conference on Computer Vision*, volume 1065, pages 709–720, 1996.

[182] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, 2010.

[183] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1768–1775, 2012.

[184] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[185] G. Teza, A. Galgaro, N. Zaltron, and R. Genevois. Terrestrial laser scanner to detect landslide displacement fields: A new approach. *International Journal of Remote Sensing*, 28(16):3425–3446, 2007.

[186] E. Thompson. A rational algebraic formulation of the problem of relative orientation. *The Photogrammetric Record*, 3(14):152–157, 1959.

[187] S. Thrun and M. Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.

[188] V. Tofani, F. Raspini, F. Catani, and N. Casagli. Persistent scatterer interferometry (psi) technique for landslide characterization and monitoring. *Remote Sensing*, 5(3):1045–1065, 2013.

[189] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[190] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372. Springer, 2000.

[191] C.-J. Tsai, N. P. Galatsanos, and A. K. Katsaggelos. Optical flow estimation from noisy data using differential techniques. In *Proceedings of the IEEE Conference on International Acoustics, Speech, and Signal Processing*, volume 6, pages 3393–3396, 1999.

[192] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundation and Trends in Computer Graphics and Vision*, 3(3):177–280, July 2008.

[193] S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[194] S. Van Huffel and J. J. Vandewalle. *The Total Least Squares Problem : Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[195] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 722–729, 1999.

[196] C. Vogel, S. Roth, and K. Schindler. An evaluation of data costs for optical flow. In J. Weickert, M. Hein, and B. Schiele, editors, *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 343–353. Springer Berlin Heidelberg, 2013.

[197] W. Wagner, A. Ullrich, V. Ducic, T. Melzer, and N. Studnicka. Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(2):100–112, 2006.

[198] C. Waldhauser, R. Hochreiter, J. Otepka, N. Pfeifer, S. Ghuffar, K. Korzeniowska, and G. Wagner. Automated classification of airborne laser scanning point clouds. In S. Koziel, L. Leifsson, and X.-S. Yang, editors, *Solving Computationally Extensive Engineering Problems: Methods and Applications*. Springer, 2014.

[199] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, 1991.

[200] S. Wang, V. Markandey, and A. Reid. Total least squares fitting spatiotemporal derivatives to smooth optical flow fields. In *Proceedings SPIE, volume 1698*, pages 42–55, 1992.

[201] A. Waxman, J. Wu, and F. Bergholm. Convected activation profiles and the measurement of visual motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 717–723, 1988.

[202] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14(1):67–81, 1995.

[203] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision*, 95(1):29–51, 2011.

[204] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer, 2009.

[205] T. Weise, B. Leibe, and L. Van Gool. Accurate and robust registration for in-hand modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[206] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *Proceedings of the British Machine Vision Conference*, volume 34, pages 1–11, 2009.

[207] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1744–1757, 2012.

[208] M. Yamamoto, P. Boulanger, J. Beraldin, and M. Rioux. Direct estimation of range flow on deformable shape from a video rate range camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):82–89, 1993.

[209] N. Yastikli and K. Jacobsen. Direct sensor orientation for large scale mapping—potential, problems, solutions. *The Photogrammetric Record*, 20(111):274–284, 2005.

[210] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007.

[211] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.