Pose Estimation and Collaborative Data Transmission in Visual Sensor Networks Equipped with RGB-D Cameras

Xiaoqin Wang

SUBMITTED IN TOTAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER SYSTEMS ENGINEERING
FACULTY OF ENGINEERING
MONASH UNIVERSITY
AUSTRALIA

2015

Copyright Notice

ABSTRACT

Low cost RGB-D cameras (or sensors) have significant potential for enhancing the performance of visual sensor network (VSN) applications. RGB-D sensors supplement the conventional red-green-blue (RGB) color information with per-pixel depth data. VSNs, when equipped with RGB-D sensors, open up possibilities for new and innovative application areas. However, to reach their full potential, they need to use their limited battery supplies very frugally, and operate autonomously. Distributed, scalable algorithms must form the backbone of a VSN system architecture.

A fundamental requirement of autonomous operation is that a VSN node needs to determine its pose (the location and orientation of its sensors), and use its communication channels as efficiently as possible. The volume of visual and depth data, generated by the sensors of a VSN, is inevitably going to be large. When sensors operate in many hostile environments (especially for disaster recovery, search and rescue operations in confined spaces), this communication problem affects the system overall performance significantly. Such critical situations present immense challenges for efficient data transmission and storage, particularly over shared wireless channels. It should also be noted that conventional localization methods such as GPS (Global Positioning System) cannot be accessible in the places where the sensors operate, such as indoor and underwater environments.

This thesis offers novel solutions to the above-mentioned problems. In the first part of the thesis, we present a solution to the sensor pose estimation problem by using color and depth information captured by each RGB-D sensor. We provide an algorithm which computes the relative pose between two sensors by matching the depth images in a distributed manner. Then, we use this algorithm together with graph theory based techniques to develop a self-calibration method which determines each sensor's pose in a network of multiple RGB-D camera nodes.

In the second part of this thesis, we address the problem of efficient data communication under bandwidth constraints. In order to reduce the VSN communication load, we provide new algorithms that allow in-node detection of redundant visual information to avoid its transmission and storage. We achieve this by determining the correlated regions in the captured imagery with the help of the sensor pose estimation methods presented in the first part of the thesis. We introduce a depth video compression scheme for a single mobile RGB-D sensor; then, we develop a collaborative color and depth data coding mechanism for multiple sensors with overlapping fields-of-view.

Experimental results obtained on an experimental VSN testbed show that the sensor pose estimation and collaborative data coding mechanism presented in this thesis is able to decrease the overall communication load by approximately 40%, leading to a 55% reduction of a sensor node's energy consumption due to a significant reduction in the number of required packet transmissions.

DECLARATION

In accordance with Monash University Doctorate Regulation 17.2: *Doctor of Philosophy and Research Master's Regulations*, the following declarations are made:

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

The core theme of the thesis is the creation of new distributed methods for sensor pose estimation and RGB-D data communication. The ideas, development and writing up of all the work in the thesis were the principal responsibility of myself, the candidate, working within the Department of Electrical and Computer Systems Engineering under the supervision of Dr Y. Ahmet Şekercioğlu (main supervisor) and Professor Tom Drummond (associate supervisor).

Signed: _____
Xiaoqin Wang

Date: August 2015

ACKNOWLEDGEMENTS

Only my name appears as writer in the front page of this PhD thesis, however this thesis would not have been possible without the support of a number of people.

First of all, I would like to express my deep gratitude to my main supervisor, Dr Y. Ahmet Şekercioğlu, for the opportunity to undertake this PhD research, for his continuous encouragement and guidance on both my work and life, for passing along his knowledge and experience, for his seemingly inexhaustible supply of interesting research problems, and for his continued patience during the write up stage of the submitted papers and this thesis.

I also owe a debt of gratitude to Prof Tom Drummond who provided invaluable advice and upheld my connection to the Australian robotics and computer vision communities. Thanks also goes to Dr Chathuranga Widanapathirana, who generously shared his own experience which helped me to quickly adapt to research life. I would like to thank Dr Dennis Lui, who guided me through the fundamental knowledge of computer vision and provided valuable suggestions during the hardest time of my research. I would like to thank Dr Alex McKnight who proofread the final draft for grammar and style.

Finally, and the most importantly, none of this would have been possible without the unwavering love and support from my family. I would like to thank my wife, Yulong. She has shared with me all the ups and downs and has never failed to believe in me. With her love and support, I can dedicate myself to research and study. A very special thank also goes to my parents, to whom I owe everything.

Contents

List of Figures

List of Tables

ABBREVIATIONS AND ACRONYMS

CABAC	Context-Adaptive Binary Arithmetic Coding
DHC-M	Differential Huffman Coding with Multiple Lookup Tables
DLT	Direct Linear Transformation
DoF	Degree of Freedom
FAST	Features from Accelerated Segment Test
FoV	Field of View
FPFH	Fast Point Feature Histograms
GoP	Group of Pictures
GPS	Global Positioning System
GPU	Graphics Processor Unit
ICP	Iterative Closest Point
IPM	Initial Pose Matrix
IW-DVC	Image Warping Based Depth Video Compression
MSE	Mean Squared Error
MV	Motion Vector
MVC	Multiple View Coding
ORB	Oriented FAST and Rotated BRIEF
PGF	Progressive Graphics File
PSNR	Peak Signal-to-Noise Ratio
RANSAC	Random Sample Consensus
RGB-D	Red, Green, Blue-Depth
RPRR	Relative Pose based Redundancy Removal
RMSE	Residual Minimum Square Error
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SSIM	Structural Similarity
SURF	Speeded Up Robust Features
UM	Uncertainty Matrix
VSN	Visual Sensor Network
WSN	Wireless Sensor Network

LIST OF VARIABLES

α_j	An element of a 6DOF motion vector.
B_b, B_f	Block coordinate set in backward check and forward prediction process of RPRR.
$B_{overall}$	Universe of B_f and B_b.
$\mathbf{C}_a, \mathbf{C}_b$	Color images captured by Sensor a and Sensor b.
D_a, D_b	Depth measurements obtained by Sensor a and Sensor b.
$\mathbf{E}$	Update transformation matrix in each iteration.
$(f_{x,a}, f_{y,a})$	Focal length of Sensor a in horizontal and vertical axes.
$\mathbf{G}_j$	6DOF motion generator matrices.
H_i, H_{i-1}	Current pixel/reference pixel with the invalid depth value.
$(i_{c,a}, j_{c,a})$	Principal point coordinates of Sensor a.
$\mathbf{M}$	Transformation matrix which describes a mobile sensor's motion in the time interval of a captured I-frame and P-frame.
$\mathbf{M}_{ab}$	Transformation matrix which describes the relative pose between Sensor a and Sensor b.
N_a, N_b	Number of sampled points on the depth images captured by Sensor a and Sensor b.
$\vec{n}_{l*,b}, \vec{n}_{k,b}$	Surface normal at point $\mathbf{p}_b^{l*}$ and $\mathbf{p}_b^k$.
$\mathbf{p}_a^l, \mathbf{p}_b^k$	Sampled points on the depth image captured by Sensor a and Sensor b.
$\mathbf{p}_a^{k*}$	Corresponding points of $\mathbf{p}_b^k$.
$\mathbf{p}_b^{l*}$	Corresponding points of $\mathbf{p}_a^l$.
$\mathbf{p}_e$	Vector representing a real world point in Euclidean space.
$q_h()$	Conditional probability distribution when the reference pixel has the invalid depth value.
$q_v()$	Conditional probability distribution when the reference pixel has a valid depth value.
S_a, S_b	Set of sample points on depth images captured by Sensor a and Sensor b
S_a^*, S_b^*	Set of corresponding points on the depth images captured by Sensor a and Sensor b.
$[u_a, v_a, 1, q_a]^T$	Inverse depth coordinates representation of a real point $\mathbf{p_e}$ in Sensor a's coordinate system.
V_i, V_{i-1}	Current pixel/reference pixel with a valid depth value.

$w_{l,a}$	Weight parameter for correspondence established between $\mathbf{p}_a^l$ and $\mathbf{p}_b^{l*}$.
$[x_a, y_a, z_a, 1]^T$	Homogeneous coordinates representation of a real point $\mathbf{p_e}$ in Sensor a's coordinate system.
X_i, X_{i-1}	Current/reference pixel.
$\mathbf{Z}_a, \mathbf{Z}_b$	Depth images captured by Sensor a and Sensor b.
$\mathbf{Z}_I, \mathbf{Z}_P$	I-frame and P-frame in a group of depth frames.

CHAPTER ONE

INTRODUCTION

1.1 Visual Sensor Networks

After decades of intensive worldwide research and development efforts, wireless sensor networks (WSNs) are becoming a mature technology. The latest advances in video technology, inexpensive camera sensors, and distributed processing allow the wide utilization of image sensors in WSNs. It has resulted in a new paradigm— visual sensor network (VSN) [SH09]. VSN is a group of networked smart cameras with image/video capturing, computing and wireless communication capabilities powered by on-board batteries. Rich information is provided on situation awareness by observing and processing image/video data. VSN is becoming increasingly popular to measure and estimate quantities of interest at spatially distributed locations. It promises a wide range of innovative applications, such as multimedia surveillance [Cuc05, PVFC13], environmental monitoring [MQC11, OS12], multimedia-aided navigation [CMMV12, YG11], industrial process control [MP11, MMLB13] and localization services [KQ11, KGS05].

1.1.1 Characteristics and Design Challenges

Compared with a single visual sensor operation, VSNs have the advantages of faster task completion, more extensive coverage, decreased vulnerabilities to sensor failures, and higher estimation accuracy through sensor fusion. Compared with conventional WSNs that provide 1D scalar data, VSNs with image sensors can

provide 2D or even 3D sets of data points. Moreover, a camera's sensing model is inherently different from the sensing model of any other types of scalar sensors. These two principal differences between VSNs and WSNs raise three major design challenges for VSNs: resource requirements, local processing, and sensor location and orientation, which are explained in the following paragraphs.

Resource Requirements

The additional dimensionality of the data set results in richer information and better awareness of the surrounding environment. Simultaneously, the additional data also lead to a higher complexity of data processing/analysis and much higher bandwidth/energy consumptions. These characteristics result in more design challenges regarding the vast visual information delivery in the networks.

The lifetime of each visual sensor depends on the on-board battery, the usage of which is proportional to the energy required by the sensing, processing and transceiving modules. The large amount of image data generated by networked sensors consumes huge amounts of energy on processing and transceiving, which are much more expensive than conventional WSNs. Furthermore, most VSN applications require the delivery of the visual content with a certain level of quality-of-service (QoS). The energy, bandwidth and processing capability constraints of the sensor nodes, as well as the nature of the wireless links that interconnect them, are much severer. Higher bandwidth and more sophisticated processing techniques are required to deliver and process the visual information in VSNs.

Local Processing

Local (on-board) processing refers to the techniques of processing the image data by each visual sensor immediately after this data has been captured. Local processing of the image data helps to reduce the overall amount of data that is required to be stored and transmitted in the network. Local processing can involve many types of lightweight image processing algorithms, such as background subtraction

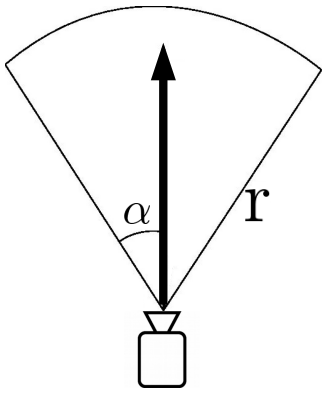


Figure 1.1: 2D directional sensing model

for motion detection [Pic04, BHH11] and feature detection for object classification [ADB+04, ABC+03, KGSL05]. Generally, depending on the various applications' requirements, visual sensors should be able to provide different levels of intelligence, which are determined by the complexity of the processing algorithms.

Sensor's Location and Orientation

In conventional WSNs, the sensing range of scalar sensors can be approximated to a round disk with a fixed radius [HT03]. The sensing and connectivity scopes of scalar sensors are equivalent and associated with the sensors' vicinity. In VSNs, as cameras capture images of distant scenes from a certain direction, the sensing range of visual sensors can be characterized as a sector, specified by the orientation and radius parameters. The maximum volume visible from a visual sensor is defined as the field of view (FoV). The depth of field (DoF) is the distance between the nearest and farthest targets in the FoV that can be observed by a visual sensor clearly [CG10]. Fig. 1.1 shows a simple graphical 2D representation of a typical visual sensor's FoV. The viewing angle is "2α" and "r" is the sensing radius.

The characteristic of directional sensing results in an important difference in localization algorithms between conventional WSNs and VSNs. In a conventional WSN, when sensors are geographically close, they sense a similar scene and have similar measurements. Therefore, WSN applications only need to know sensor

location information. However, in a VSN, sensors which are geographically close may not sense a similar scene due to orientation differences or visual occlusions. Therefore, in addition to sensor location information, each sensor's orientation information must be acquired by VSN applications.

The combination of camera's location and orientation is referred to as the pose of the camera. This information can always be acquired through a camera calibration process, which retrieves the camera's intrinsic and extrinsic parameters. Estimation of calibration parameters usually requires correspondences between multiple sets of patterns extracted from the images captured by different cameras. In order to establish the correspondences, some algorithms require specially designed calibration patterns to be visible in all images, or the precise pose information of calibration patterns/objects have to be known [CDS00, Zha00]. Some other algorithms using the visual features do not require special calibration patterns and can match feature points between different images [LF06, BBD14]. Based on the camera calibration parameters, a deployment map with the sensor's location and orientation information in the VSN can be established.

1.1.2 RGB-D Camera-Equipped Visual Sensor Networks

With the invention of low-cost RGB-D cameras such as the Microsoft Kinect [FSMA10] , high-resolution depth and visual (RGB) sensing has become available for widespread use. Kinect was initially used as an input device by Microsoft for the Xbox game console [kin]. With Kinect, people are able to interact with games without the need to touch a controller. With Kinect's wide availability and much lower cost than traditional 3D cameras (such as stereo cameras [SL04] and time-of-flight (TOF) cameras [FAT11]), many researchers and practitioners in computer vision, multimedia, and robotics communities have discovered that the depth sensing technology of Kinect can be extended far beyond the gaming industry. Furthermore, the complementary nature of the depth and visual information gives Kinect the potential to find new solutions for classical problems in

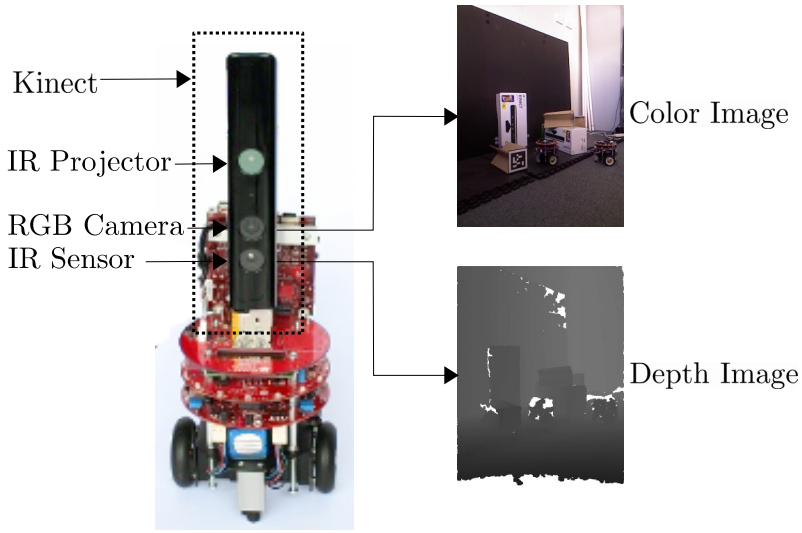


Figure 1.2: A Microsoft Kinect RGB-D camera, shown installed vertically on Monash University's experimental mobile robot "eyeBug" to construct a mobile RGB-D sensor. Samples of captured color and depth images are also shown.

various research fields, including three-dimensional (3D) mapping and localization [HKH+12, EHS+14], or scene/object reconstruction [NDI+11, KK13].

Microsoft Kinect (shown installed on WSRNLab's [wsr] experimental robot eyeBug in Fig. 1.2), contains an RGB camera and an IR projector-camera pair, which can produce color images with per-pixel depth information at a rate of 30 fps. As described by its inventors [FSMA07], depth is measured via a triangulation process that is based on the detection of transverse shifts of local dot patterns in the IR speckle with respect to its reference patterns at a known distance to the device. This process is repeated for all local regions in the IR speckle and produces a disparity-based depth image. The default RGB video stream provided by Kinect is in 8-bit VGA resolution (640×480 pixels) and the monochrome depth video stream is also in VGA resolution with 11-bit depth, which provides 2,048 levels of sensitivity.

Implementing the low-cost, small-size Kinect sensor in VSNs makes it possible to collect depth data distributively in cost-effective ways. RGB-D camera-equipped

VSNs, by using the additional depth data, can significantly enhance the performance of conventional applications, such as immersive telepresence or mapping [SEE+12, TZL+12], environment surveillance [CPS11, LXW+12], or object recognition and tracking [AJ13, AZD13], and provide possibilities for new and innovative applications [RYMZ13, WLC15]. The value of VSN applications is even more important, especially in places that humans are not able to access, such as search and rescue operations after earthquakes or nuclear accidents. An illustrative scenario is shown in Fig. 1.3. On the other hand, RGB-D sensors inevitably generate vast amounts of visual and depth data. The extra depth information provides opportunities to determine sensor pose in different approaches from the calibration methods for conventional camera sensors. However, the huge amount of color and depth data results in very severe constraints with limited resources. The energy, bandwidth, and processing capability constraints of the sensor nodes still exist and are much severer. *Therefore, new methods which consider the characteristics of depth sensing are urgently required for VSNs equipped with RGB-D cameras.*

1.2 Thesis Objective and Contributions

To summarize, this thesis investigates two major challenges posed by RGB-D camera-equipped VSNs: (1) RGB-D sensor pose estimation, and (2) RGB-D data communication. Several algorithms have been proposed to accurately estimate sensor pose information and achieve efficient color and depth image communication over the network in indoor scenarios. The main contributions of this thesis are presented below and shown in Fig. 1.4.

1.2.1 RGB-D Sensor Pose Estimation

In VSNs, visual sensor pose estimation is a prerequisite to accomplish a wide range of collaborative tasks. Therefore, this thesis first focuses on estimating RGB-D sensor location and orientation.

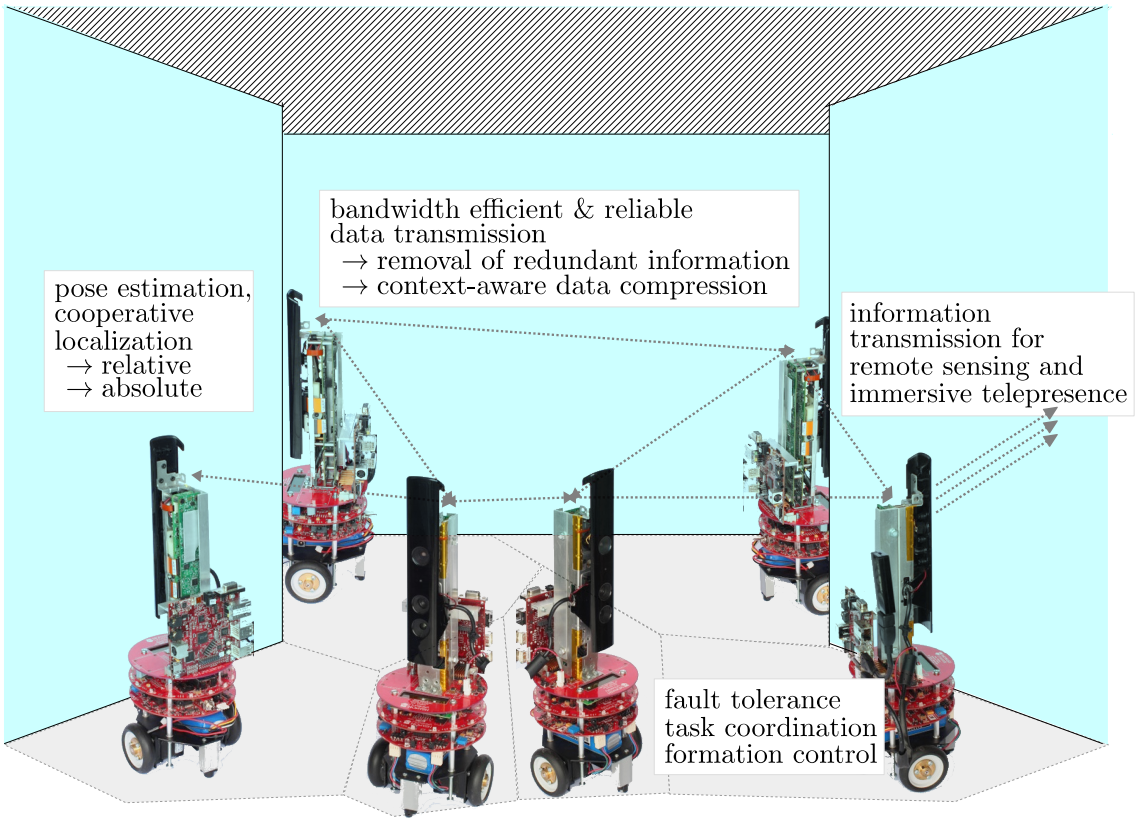


Figure 1.3: An indoor mapping and exploration scenario showing the Monash University's RGB-D camera equipped experimental mobile sensors "eyeBugs" [DLL+11, eye11]. A typical application would be mapping indoors after a disaster such as the Fukushima nuclear reactor accident. As shown in the diagram, there are numerous challenges that need to be addressed. In this thesis, we tackle the sensor pose estimation and bandwidth efficient data transmission problems.

1. **Depth Image Registration for Relative Pose Estimation**

   The first important contribution of this thesis is an algorithm which estimates two RGB-D sensors' relative poses by determining the correlation in overlapping sensor observations. It is a peer-to-peer, distributed depth image registration algorithm, estimating the relative pose between multiple sensors when sensors observe a common scene from different viewpoints.

   In this algorithm, a maximum likelihood framework based on the beam-based sensor model [TBF05] is devised and incorporated with the Iterative Closest Point (ICP) algorithm to enhance the limited performance of ICP variants in relative pose estimation. The proposed framework can eliminate the adverse effects of the situations where two views of a scene each are partially seen

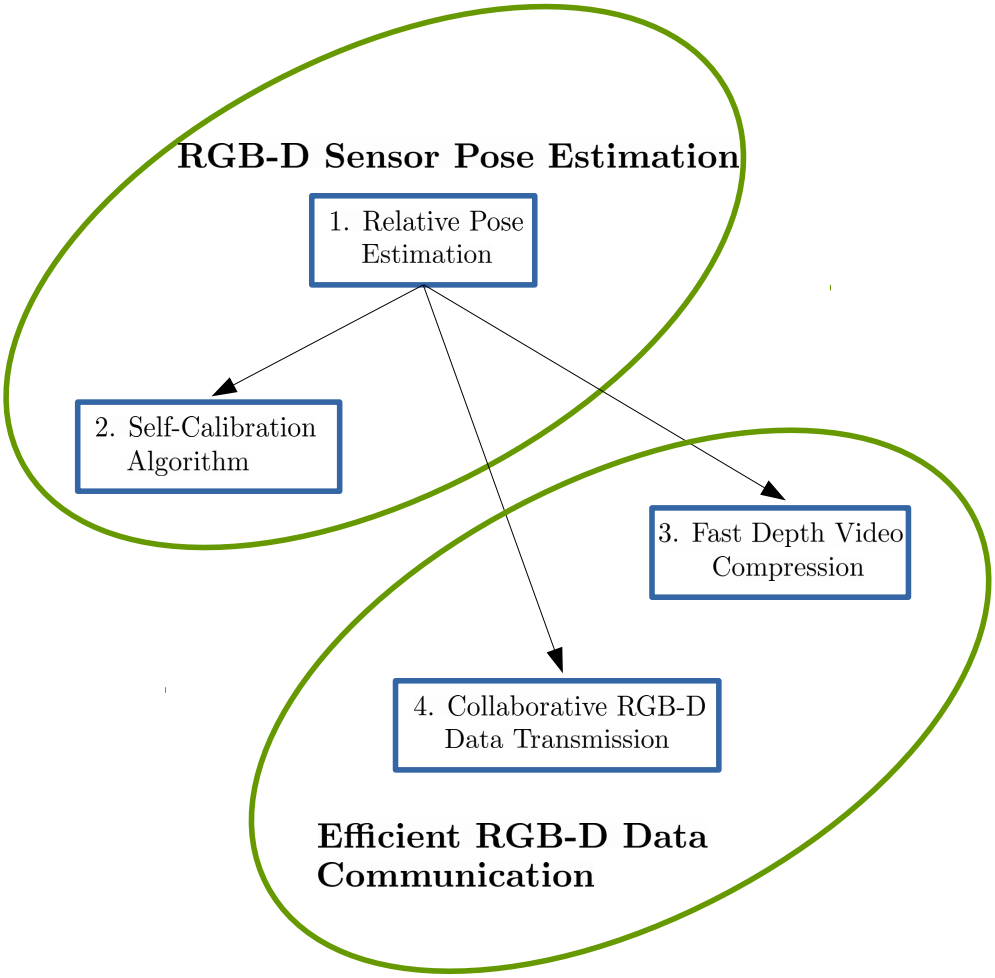


Figure 1.4: Contributions of the project.

by the sensors. Then, in order to cancel the bias introduced by the beam-based sensor model, we have developed a scheme that allows the algorithm to symmetrize across the two views. The algorithm was implemented and tested both on a laptop and our visual senor network testbed comprised of mobile RGB-D sensors.

2. **Self-Calibration Algorithms**

The second important contribution of this thesis is a self-calibration algorithm which determines each sensor's pose globally in the network. We first model a visual sensor network as an edge-weighted graph. The graph represents FoVs of sensors as vertices and overlapping FoVs as edges, respectively.

Then, based on this model, and by using the real-time color and depth data, the sensors with shared FoVs estimate their relative poses in pairwise. Our approach assumes that each sensor in the network has overlapping FoV with at least one another sensor and the visual sensors have been internally calibrated prior to deployment. The system does not need the existence of a single common view shared by all sensors, and it works in 3D scenes without any specific calibration pattern or landmark. Since the proposed scheme distributes working loads evenly in the system, they are scalable and the computing power of the participating sensors is used efficiently.

1.2.2 Efficient RGB-D Data Communication Schemes

This thesis then focuses on removing the redundancy in the captured information to realize efficient communication in the network. In order to achieve this goal, we propose two algorithms which can explore the correlation in observations and prevent the redundant information from being transmitted in two circumstances: (1) depth video captured by a mobile sensor, and (2) color and depth images obtained by multiple sensors with overlapping FoVs, respectively.

1. **Fast Depth Video Compression**

   The third important contribution of this thesis is an algorithm, named 3D Image Warping Based Depth Video Compression (IW-DVC), for fast and efficient compression of depth images captured by mobile RGB-D sensors. We have designed the IW-DVC method to exploit the special properties of the depth data and achieve a high compression ratio while preserving the quality of the captured depth images. Our solution combines sensor pose estimation with 3D image warping techniques, and includes a lossless coding scheme which is capable of adapting to depth data with a high dynamic range. IW-DVC operates at high speed, is suitable for real-time applications, and is able to attain an enhanced motion compensation accuracy compared

with conventional approaches. In addition, it removes the existing redundant information between the depth frames to further increase compression efficiency.

2. **Collaborative RGB-D Data Transmission**

   The fourth important contribution of this thesis is a collaborative transmission algorithm. As the same scenery may be observed by multiple sensors, collected images will inevitably contain significant amounts of correlated information, and transmission load will be unnecessarily high if all the captured data are sent. We focus on this issue, and present a novel approach in developing a comprehensive solution for minimizing the transmission of redundant RGB-D data in VSNs. Our framework, called Relative Pose based Redundancy Removal (RPRR), is based on relative pose estimation between pairs of RGB-D sensors [WŞD13a] and 3D image warping techniques [Feh04] to locally determine the color and depth information, which can only be seen by one sensor but not the others. Consequently, each sensor is required to transmit only the uncorrelated information to the remote station. In order to further reduce the amount of information before transmission, we apply a conventional coding scheme based on discrete wavelet transform [AH92] with progressive coding features for color images, and a novel lossless differential entropy coding scheme for depth images. In addition, at the remote monitoring station, we use post-processing algorithms created by us to deal with the artifacts that could occur in the reconstructed images due to the under-sampling problem [Mar99].

1.3 Organization of the Thesis

The thesis is organized as follows:

Chapter 2: Literature Review surveys the relevant literature. As solving the sensor localization issue is the prerequisite for collaborative tasks in sensor

networks, this chapter first explains the key issues in RGB-D sensor pose estimation and self-calibration in VSNs. The chapter then examines the approaches used for color and depth image communication in VSNs. This chapter provides a foundation for the research problems and state-of-the-art solutions.

Chapter 3: Relative Pose Estimation Between Two RGB-D Sensors investigates the relative pose estimation scheme for multiple RGB-D sensors. This chapter explains the characteristics of the depth information returned by RGB-D sensors and provides the mathematics model of the 6 degrees of freedom (DoF) relative pose. A distributive algorithm is proposed, which uses depth image registration to estimate the relative pose. A mobile visual sensor network testbed, consisting of two RGB-D sensors, is also constructed.

Chapter 4: Self-Calibration for RGB-D Camera-Equipped VSNs investigates sensor pose estimation problems in VSNs with more than three RGB-D sensors. A self-calibration algorithm is presented, which models the network as an edge-weighted graph and transfers the self-calibration to the shortest path problem. In this algorithm, feature detection/matching and basic graph theory methods are adopted.

Chapter 5: Efficient RGB-D Data Communication Schemes addresses the efficient color and depth image communication problem in bandwidth limited situations. This chapter carefully considers the RGB-D data communication in two situations: depth video captured by a mobile sensor and multi-view color/depth images captured by multiple static sensors. Two frameworks are proposed for different situations. The proposed schemes use the algorithm developed in Chapter 3 to estimate the mobile sensor's motion and multiple static sensors' relative poses. The correlation in the observed information can be explored and the redundancy can be removed by using the pose information.

Chapter 6: Conclusions and Future Works summarizes the main results and contributions of this thesis. It also points out some future works that can be undertaken in this area.

1.4 Publications

During the course of this project, a number of publications based on the work presented in this thesis have been produced. They are listed here for reference:

1.4.1 Journal Articles

- X. Wang, Y. A. Şekercioğlu, T. Drummond, "Vision-Based Cooperative Pose Estimation for Localization in Multi-Robot Systems Equipped with RGB-D Cameras", *Robotics*, vol. 4, pages: 1-22, January 2015. [WŞD14].
- X. Wang, Y. A. Şekercioğlu, T. Drummond, E. Natalizio, I. Fantoni, and V. Fremont, "Fast Depth Video Compression for Mobile RGB-D Sensors", *IEEE Transactions on Circuits and Systems for Video Technology*, 2015 [WŞD$^+$15b]. .
- X. Wang, Y. A. Şekercioğlu, T. Drummond, E. Natalizio, and I. Fantoni, "Relative Pose Based Redundancy Removal: Collaborative RGB-D Data Transmission in a Mobile Visual Sensor Network", *IEEE Transactions on Multimedia* (under review) .

1.4.2 Conference Papers

- X. Wang, Y. A. Şekercioğlu, T. Drummond, "Self-Calibration in Visual Sensor Networks Equipped with RGB-D Cameras" in *Proceedings of 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Pages: 2289-2293, Brisbane, Australia, April, 2015. [WŞD15a].
- X. Wang, Y. A. Şekercioğlu, T. Drummond, "A Real-Time Distributed Relative Pose Estimation Algorithm for RGB-D Camera Equipped Visual Sensor

Networks" in *Proceedings of Seventh ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)*, Pages: 68-74, Palm Springs, USA, October, 2013. [WŞD13a].

- X. Wang, Y. A. Şekercioğlu, T. Drummond, "Multiview Image Compression and Transmission Techniques in Wireless Multimedia Sensor Networks: A Survey" in *Proceedings of Seventh ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)*, Pages: 258-265, Palm Springs, USA, October, 2013. [WŞD13b].

- X. Wang, Y. A. Şekercioğlu, T. Drummond, "PhD Forum: An Efficient Communication Scheme for Mobile Visual Sensor Networks Equipped with RGB-D Cameras" in *Proceedings of Seventh ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)*, Pages: 278-279, Palm Springs, USA, October, 2013. [WŞD13c].

CHAPTER TWO

SENSOR POSE ESTIMATION AND RGB-D DATA COMMUNICATION: THE CURRENT STATE-OF-THE-ART

Sensor localization and efficient data communication schemes need to be designed for VSNs equipped with RGB-D cameras. In this chapter, we review the state-of-the-art research related to visual sensor pose estimation and efficient color/depth data communication. For the sensor pose estimation problem, we first study the methods which determine RGB-D sensor motion. Then, we review the calibration algorithms in detail to estimate multiple visual sensor poses in VSNs. For the efficient color and depth data communication issue, we first evaluate depth video coding schemes, then present a comprehensive survey of multi-view image communication in VSNs.

2.1 Depth Image Registration for RGB-D Sensor Pose/Motion Estimation

RGB-D cameras can provide color images as well as depth images at a high frequency. The pixels in a depth image contain the range information between the observed scene and the sensor. Using the relation between the pixel value and range information, a 3D point cloud can be extracted from each depth image. Consider a situation where two RGB-D sensors observe the same scene from different viewpoints. By matching two point clouds which are extracted from the corre-

sponding depth images, the transformation between them and, consequently, the 6 DoF relative pose between two sensors can be deduced.

This section reviews the registration methods for depth images captured by RGB-D sensors. The state-of-the-art studies are classified into three main categories: (1) Iterative Closest Point (ICP) variants, (2) feature-based registration, (3) hybrid approaches. After reviewing of the approaches, a detailed comparison is presented.

2.1.1 Feature-Based Registration

Instead of matching two 3D point clouds directly, feature-based registration methods, which try to reduce the number of points from both point clouds, use detection and feature descriptors to represent the input data. A feature includes a position in the image coordinate and a descriptor which contains the information around the feature position. Feature points can be detected from not only color images (visual features) but also depth images (3D features). Irrespective of the kind of feature used, these approaches operate as follows: in the first step, feature points can be detected by feature detection algorithms. In the second step, by matching the features and evaluating the depth images at the locations of these feature points, a set of point-wise 3D correspondences between two frames is obtained. In the last step, these point pairs are used to compute the transformation. These steps are shown in Fig. 2.1.

There are various feature detectors and descriptors which deal with 2D visual data [TM08] (e.g. scale-invariant feature transform (SIFT) [Low99], speeded up robust features (SURF) [BETVG08], features from accelerated segment test (FAST) [RD06a]). These algorithms are able to identify feature points among disordered data with a descriptor invariant to uniform scaling, orientation, and partially invariant to distortion and illumination changes. Only a few 3D feature detectors/descriptors have been proposed. A 3D descriptor, named fast point feature histograms (FPFH) [RBB09], is based on a histogram of the differences of angles between the normals of the neighboring points of the source point. Some exten-

sions of the 2D Harris detector [HS88] are proposed in [G09], which can be used as detector and descriptor for 3D features. One of the most common methods for finding the transformation between correspondences is Random Sample Consensus (RANSAC) [FB81]. The aim of the RANSAC algorithm is to determine a suitable model which estimates the position transformation best. At each iteration of the algorithm, a number of correspondences are randomly selected and considered as inliers. A transformation model is determined to fit these correspondences. The remaining correspondences are then tested against the fitted transformation and included as inliers if their error is within a given threshold. This process is iterated a number of times until the best solution is determined. Correct correspondence between the pixels in the RGB image and the pixels in the depth image is the prerequisite for this type of approach.

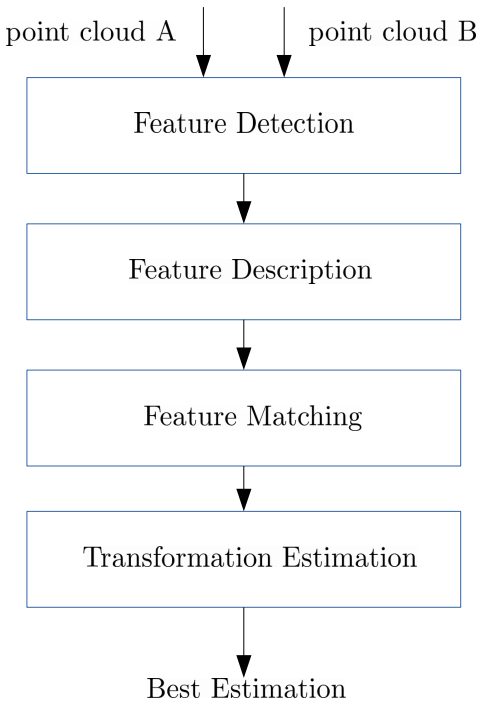


Figure 2.1: General steps of feature-based registration.

Zou et al. [ZCWL12] follow the above steps to estimate the RGB-D sensor's pose and evaluate the performance of different feature detection algorithms. Instead of making a Kinect orient horizontally, Wang et al. [WML+12] make the Kinect look up and focus on matching the features on the ceilings. After the odometry results

are obtained through feature matching, they feed the results into the Gmapping algorithm [GSB07]. A Hokuyo Laser Range Finder [KTCS09] is also used to obtain range measurements from the surroundings and produce a map of the environment. In addition to these conventional feature detection approaches, Zeisl *et al.* [ZKP13] use orthographic projection of RGB-D data to simplify matching. This algorithm exploits characteristic of the salient directions in the scene, which are repeatable in different scans. Each salient direction is then exploited to render an orthographic view.

2.1.2 ICP Variants

The ICP algorithm [BM92] was developed to register two point clouds by computing the rigid transformation between them. Different from feature-based registration, ICP and its variants commonly use all the available depth points directly instead of using the feature points. It became popular following its successful application in the registration of highly accurate range data from laser rangefinders. A thorough survey is available in [SMFF07]. To estimate RGB-D sensor pose, ICP operates by iteratively matching point clouds extracted from time-adjacent depth frames to converge upon an estimated sensor pose change which describes the points' movement. Distance information between matched points is commonly used to compute the transformation which best explains the alignment of two point clouds. At each iteration the algorithm attempts to find an update to the transformation that minimizes a cost function, the error metrics of which are defined based on the point-to-point [GIRL03], point-to-plane [CM91] or other geometrical relationships [PS03]. The general steps of the ICP framework are shown in Fig. 2.2.

The ICP starts with two point clouds A & B and an initial transformation. Then ICP iterates in four consecutive steps. Firstly, the initial transformation is applied on point cloud A. Secondly, the corresponding point pairs are established between transformed point cloud A and point cloud B. Thirdly, by using the correspondence information, the transformation which minimize error metric are

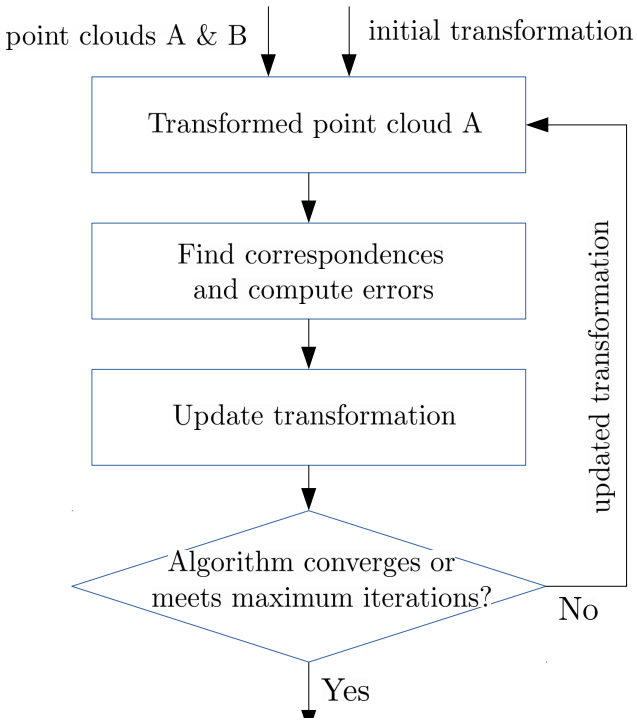


Figure 2.2: General steps of ICP.

computed. Finally, the framework checks the threshold to determine whether the algorithm converges. The above process operates iteratively until the algorithm converges. Once the algorithm converges, the registration is considered completed and the transformation is determined.

As a typical RGB-D frame has hundreds of thousands of points, performing ICP on the full point cloud is computationally expensive. In order to alleviate this problem, a common method is to subsample the data to speed up operations at the cost of accuracy. This describes a fundamental trade-off in the performance of ICP: registering using dense point clouds yields more accurate alignment, however it is done with lower processing frame rates. Registering a small number of the subsampled point clouds results in lower accuracy, but a higher frame rate. If frames are processed at a low frequency, the translation and rotation of the camera between two processed frames can be large, leading to a convergence failure in the algorithm. Therefore, for the best ICP variant performance, a careful balance between data size and processing frequency is required.

A number of ICP variants have been developed recently for RGB-D sensors.

Lui et al. [LTDL12] propose a fast variant of ICP which samples only a proportion of the points on each depth frame and uses inverse depth coordinates instead of Euclidean coordinates to align range data provided by a Kinect. This variant of ICP runs at an average of 28 frames per second without the help of a Graphics Processing Unit (GPU) and is robust to noise and outliers due to the use of robust estimators in an iterative reweighted least squares framework. In their experiments, they found the point-to-plane metric is much better than the point-to-point metric at coping with large inter-frame motion while remaining accurate and maintaining real-time performance. The convergence threshold for correct registration of two point clouds is also determined, which can be used for fast egomotion estimation in real time.

KinectFusion [NDI+11, IKH+11] introduces another ICP variant, which builds a scene model of the observed environment and computes the pose of the camera simultaneously. The scene model is represented as a volumetric truncated signed distance function. Each point in the 3D world is stored as the distance against the closest surface (positive for points outside the surface and negative for points inside the surface.) and some weight values. The registration in KinectFusion, which is performed between frames and the model instead of consecutive frames, allows the system to avoid accumulation error and obtain smooth camera trajectories. KinectFusion improves the accuracy of registration by operating ICP on a full point cloud generated from each depth frame. In this system, projective data association [RL01] is used to find the corresponding points along the ray (i.e. projected onto the same image coordinates). Finally, the compatibility of corresponding points is tested to reject outliers, based on distance and surface normal difference thresholds. As full point cloud is adopted in this approach, the state-of-the-art GPU is implemented to achieve online processing.

A two-stage registration approach which attempts to address the trade-off between speed and accuracy is proposed in [DJXay]. In the first stage, edge detection is adopted to determine edge features in the RGB images. Once the edge features

are extracted from two RGB-D frames, points in the corresponding depth frames are incrementally sampled on these edges and matched using ICP to find a rough transformation. In the second stage, the rough transformation is treated as the initial guess for performing ICP on the full point cloud data. However, edge detection and large amounts of point data restrict the whole process to operate below 4Hz.

2.1.3 Hybrid Approaches

As the initial guess significantly affects the accuracy of the ICP variants and ICP is commonly used to refine a nearly close registration, some researchers have combined ICP variants and feature-based registration algorithms to improve registration accuracy. These algorithms first use feature-based registration to find a coarse transformation. Then, the estimated coarse transformation is used as the initial guess in ICP frameworks.

Takeda et al. [TAT+12] propose a hybrid algorithm to achieve the self-localization of a Kinect sensor, in which ICP is used to determine the transformation between two sets of feature points. In this algorithm, feature points are detected from two consecutive color images by SURF. The outliers are deleted using the Smirnov-Grubbs test [Gru50]. The corresponding depth information of the detected feature points are found from the depth frames. Then ICP is adopted to determine the transformation which can best describe the feature points movement.

Henry et al. [HKH+12] introduce RGB-D mapping, in which egomotion estimation is achieved by alignment between RGB-D frames. It extracts feature points from the RGB images using FAST features and Calonder feature descriptors [CLF08], and matches them via the RANSAC procedure. The resulting feature matches are then combined with dense ICP acting as an initialization to determine the best alignment between consecutive frames.

In addition to using SIFT and SURF to extract the features and using ICP to estimate the relative motion between two consecutive camera positions, [HTL12] proposes a ”keyframe” concept to minimize the overall accumulation error during

sensor motion estimation. This technique uses a certain frame as a reference frame for computing the relative pose, which can avoid the rapid error accumulation in every two consecutive frames. Moreover, this method is faster, since fewer frames are used to extract the feature points.

Unlike most combinational approaches using feature-based registration to provide the initial guess for ICP, [Sch12] proposes an algorithm switching between feature-based registration and ICP. The proposed algorithm works in the beginning like feature points registration flow, but performs a refinement with the ICP in the alignment pose estimation phase. In the case that not enough homologous key point pairs can be found, the algorithm immediately switches to the ICP process.

Dryanovski et. al [DMKX12] present a technique extracting the color/depth feature and using ICP to match the feature points. Different from the approaches introduced above that only consider the feature points in color images, in this approach the feature extraction involves extracting edges from a brightness image, extracting edges from depth variation in the range data, and extracting normal vector-based edge features. The normal vectors are extracted from range data locally for each feature point and the noise is eliminated using a mode histogram morphology technique. This process makes it relatively easy to extract the edges by applying a set of morphological operations. All three types of edge features complement each other and hence are useful for accurate registration of point clouds. The algorithm achieves a 10 Hz update rate running on a normal desktop.

2.1.4 Limitations

Figure 2.3 outlines the key steps in the three classes of pose estimation approaches. All of the reviewed approaches have disadvantages and limitations, as follows:

- *ICP variants:* highly cluttered and heavily occluded scenes pose a huge problem to ICP-based algorithms due to the large differences between two depth frames. Furthermore, large inter-frame motion, which leads to a limited overlapping region between two frames, may cause the algorithms to fail.

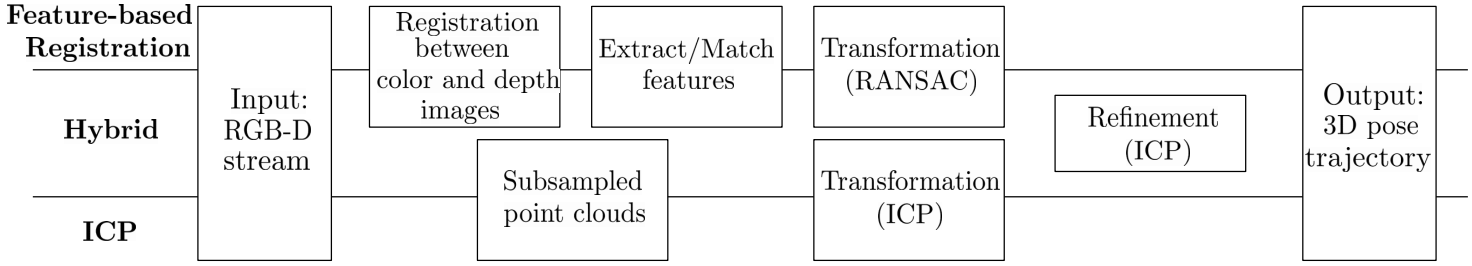


Figure 2.3: Overview of application flow of feature-based registration, hybrid and ICP approaches.

- *Feature-based registrations:* (1) computationally heavy, (2) insufficient and incorrect correspondences due to scenes without sufficient visual content and consistent illumination, (3) low registration accuracy. The first limitation constraining the frame rate means the system can hardly operate in real time. The second limitation raised by textureless environments or regions with repeated patterns affects the robustness of the algorithm and may lead to the failure of the whole system.

- *Hybrid approaches:* although the approaches combining ICP variants and feature registration algorithms are more accurate, they have the disadvantages and limitations of ICP and feature-based registration. One of the most significant disadvantages is that hybrid approaches are time-consuming and require GPUs to operate in real-time.

The disadvantages of these algorithms prevent them from being directly applied to estimate the relative pose between sensors with limited computational ability. As a result, novel algorithms which are able to estimate the relative pose between multiple sensors in real-time need to be developed.

2.2 Extrinsic Calibration for VSNs

The use of low-cost visual sensors in sensor network applications is imminent. Small low-power sensors offer an information-rich sensing modality that can detect features from a scene, perform visual confirmation, and complement other sensing modalities. To improve the sensing quality, a certain level of collaboration among sensor nodes is required. In order to accomplish collaborative tasks, an important prerequisite is that sensors have knowledge of the other sensors' location and orientation information. Namely, each camera has to know how it is oriented and other cameras' orientations, at any instant, with respect to a certain common reference frame. The importance of this is clear: assume that an external agent, which has to be tracked, is exiting from the range of the i^{th} camera and entering

that of the j^{th} one. In this case, camera i has to communicate to camera j to move and follow the agent before camera i loses it. Clearly, both cameras must share the same reference frame.

A common solution to achieve this goal is to perform extrinsic calibration in VSNs. In this section, we first study the pinhole camera model with intrinsic and extrinsic parameters. Then, we thoroughly review the state-of-the-art extrinsic calibration approaches. As the use of RGB-D sensors in VSNs has not yet become ubiquitous, no studies have been published on calibrating RGB-D camera-equipped VSNs. Therefore, we review the most relevant works to this topic, which is calibrating a VSNs with conventional cameras.

2.2.1 Pinhole Camera Model

The image acquisition process known as the *pinhole camera model* [HZ04], defines the geometric relationship between a 3D real world point and its 2D corresponding projection onto the image plane of an ideal pinhole camera. In the pinhole camera model, rays from the scene are projected onto a planar screen after transmitting through a pinhole. The screen is defined as the image plane of the camera. The center of the perspective projection (the point at which all the rays intersect) is denoted as the optical center or camera center, and the intersection point of the image plane with the optical axis is called the principal point. A pinhole camera which models a perspective projection of 3D points onto the image plane is illustrated in Fig. 2.4.

A. Intrinsic Parameters

In a camera with its pinhole located at the origin of a Euclidean coordinate system, f is the focal length of the camera and image plane aligns at $Z = f$. According to this setup, a scene point $\mathbf{P}$ with coordinates $(X_p, Y_p, Z_p)^T$ projects onto image plane

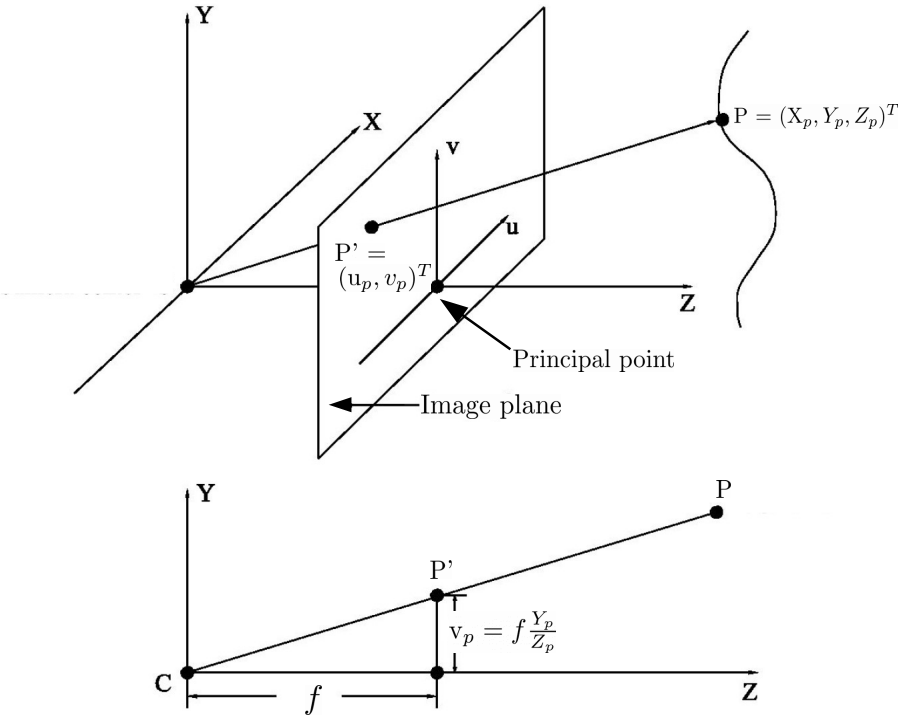


Figure 2.4: The ideal pinhole camera model indicates the relationship between a 3D point $(X_p, Y_p, Z_p)^T$ and its corresponding 2D projection $(u_p, v_p)^T$ onto the image plane.

point P' with coordinates $(u_p, v_p)^T$, such that

$$u_p = f\frac{X_p}{Z_p}, v_p = f\frac{Y_p}{Z_p}. \tag{2.1}$$

This relation can be expressed in matrix notation as,

$$\lambda \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix}, \tag{2.2}$$

where $\lambda = Z$ is the homogeneous scaling factor. Intuitively, the pinhole imaging model mapping a 3D scene to a 2D plane performs a many-to-one matching and is irreversible.

In practice, the origin of the 2D image coordinate system does not coincide with where the Z axis intersects the image plane. Therefore, We need to translate P' to

the desired origin. Let the translation (*principal point offset*) be defined as (o_u, o_v). Then, the coordinate of 2D point P' is

$$u_p = f\frac{X_p}{Z_p} + o_u, v_p = f\frac{Y_p}{Z_p} + o_v. \tag{2.3}$$

Using homogeneous coordinates, the principal-point position can be readily integrated into the projection matrix. The perspective projection equation becomes

$$\lambda \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & o_u & 0 \\ 0 & f & o_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix}. \tag{2.4}$$

In Equation 2.4, $\mathbf{P}$ is expressed in length metrics, such as meters and centimeters. As the projection is in the image plane, $\mathbf{P'}$ is expressed in pixels. Therefore, in order to find $\mathbf{P'}$ in the image plane, the resolution of the camera in pixels/meters needs to be determined. To derive the relation described by Equation 2.4, it was implicitly assumed that the pixels are square and the resolution will be identical in both u and v directions of the image plane. However, both assumptions may not always be valid. Therefore, for a more general case, the imperfections of the imaging system can be taken into account in the camera model, using the parameters m_u and m_v as the pixel scales in u and v directions. τ is used to model the skew of the pixels. The projection mapping can now be updated as

$$\lambda \begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \begin{bmatrix} m_u f & \tau & o_u & 0 \\ 0 & m_v f & o_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = [\mathbf{K}\ \mathbf{0_3}]\mathbf{P}, \tag{2.5}$$

in which $\mathbf{P} = (X_p, Y_p, Z_p, 1)^T$ being a 3D point defined with homogeneous coordinates. The intrinsic parameters of a camera are denoted as $\mathbf{K}$, which includes focal

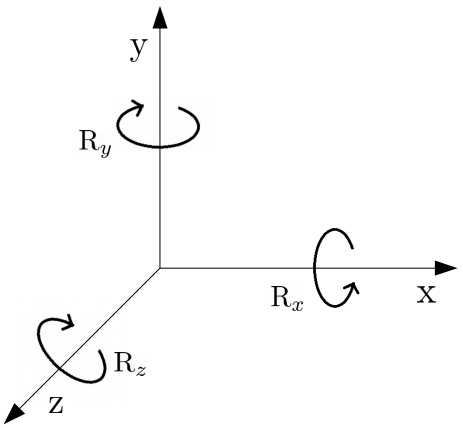


Figure 2.5: Rotation using matrices.

length, pixel scales, skew coefficient, and principal point offset.

B. Extrinsic Parameters

Different from the intrinsic parameters which describe the internal parameters of the camera, the extrinsic parameters indicate the coordinate system transformations from 3D world coordinates to 3D camera coordinates. Similarly, the extrinsic parameters define the external position and orientation of the camera in the 3D real world, and the position and orientation of the camera can be defined by a 3×1 vector **T** and by a 3×3 rotation matrix **R**. **T** is the position of the origin of the world coordinate system expressed in the coordinates of the camera-centered coordinate system.

In a standard Cartesian 3-D coordinate system, three basic rotation matrices are defined. Each rotation matrix indicates the rotation around each of the three axes, as shown in Fig. 2.5. For rotations of angles α, β, and γ around the x, y, and z axis respectively, the rotation matrices are

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{2.6}$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{2.7}$$

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

The rotation matrix $\mathbf{R}$ can be simply obtained from these three using matrix multiplication, $\mathbf{R} = \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)$. Since matrix multiplication is not commutative, the order of multiplication is important.

2.2.2 Extrinsic Calibration for Multiple Cameras

Camera calibration involves the estimation of both intrinsic and extrinsic camera parameters. The method for calibrating a single camera's intrinsic parameters is well-recognized and fixed. These parameters are obtained by using a calibration rig with known geometry, usually a checkerboard pattern [HZ04]. By capturing various perspective views of the checkerboard, the algorithm estimates the intrinsic parameters of the camera. Compared with calibration approaches for intrinsic parameters, the methods for multi-camera extrinsic calibration are much more various. Fundamentally, we can classify these techniques into two categories:

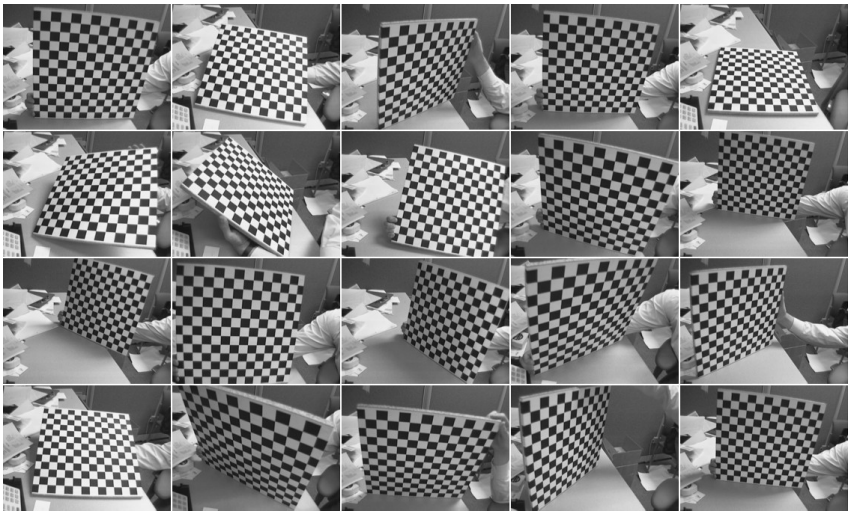
manual calibration and self-calibration. In this section, we focus on reviewing the approaches for extrinsic calibration in VSNs or multi-camera systems.

A. Manual Calibration

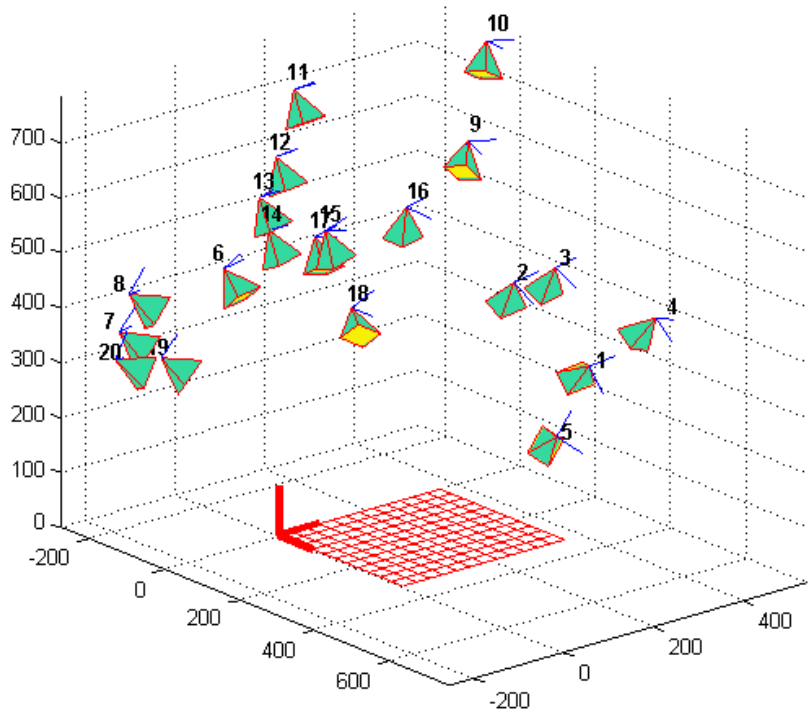
Manual calibration approaches require a particular calibration object and user interaction. A special calibration object is required to be visible in all images, or the precise pose information of calibration patterns/objects have to be known [KSA+01]. The calibration pattern is usually a 3D, 2D (planar), or virtual calibration object of precisely known geometry. The calibration object usually consists of two or three planes orthogonal to each other [Tsa87]. Sometimes, a plane undergoing a precisely known translation is also used [Zha00]. The constraint of these approaches is that they require the calibration pattern to be observed by all the visual sensors in the networks.

One of the most popular approaches is proposed by Zhang [Zha00]. This method is initially designed to calibrate a single camera's intrinsic and extrinsic parameters. This approach can be extended to accomplish the extrinsic calibration of multi-camera systems in which the intrinsic parameters of the cameras are calibrated in advance. In this approach, a chessboard pattern is first printed and attached to a planar surface. Secondly, this pattern is placed at a place which can be observed by all the cameras. Then, the extrinsic calibration of multiple cameras is achieved by establishing the correspondences across different views and solving a particular homogeneous linear system which captures the homographic relationships between multiple perspective views of the same pattern. This calibration approach is widely used because it is more natural to capture multiple views of a single planar surface - like a chessboard - than to construct a precise 3D calibration rig, as required by the Direct Linear Transformation(DLT) method [HZ04] in practice. Fig. 2.6 demonstrates a practical application of multi-plane camera calibration from multiple views of a chessboard.

Some easily detectable single features which require human interaction, such



(a) Multiple views of a chessboard



(b) Estimated locations and orientations

Figure 2.6: Extrinsic calibration example for 20 views [Zha00].

as moving a LED in a dark room, can also be used to manually calibrate multiple cameras [CDS00, SHG02, SPMP05]. As proposed by Chen et al. [CDS00], users are required to wave an identifiable point in front of all cameras. After each camera's intrinsic parameters are calibrated separately, the method generates a rough estimate of camera pose by operating pair-wise structure-from-motion on the observed points. Then, the pair-wise registrations are combined into the same coordinate frame. By using the initial camera pose, the moving point can be tracked in 3D world. The route of the point is adopted as a "virtual calibration object", which is used to improve the accuracy of the initial estimate of the camera pose. The above process is performed iteratively, which enhances the precision of the extrinsic parameters.

Svoboda et al. [SHG02] presented an approach in which a person moves a laser pointer around the scene. The very bright projections of the laser must be detected in each image with sub-pixel precision by fitting an appropriate point spread function. These detected projections are then accumulated over time to generate a virtual 3D object. Then, this virtual object which can be observed by all cameras in the system, is used to estimate each camera's orientation. This implementation of the factorization method requires correspondences across all images.

From a practical viewpoint, manual calibration methods, although they provide accurate results, require special equipment or time-consuming manual measurements and are therefore not appealing.

B. Self-Calibration

Compared to manual calibration approaches, self-calibration methods are more flexible. Self-calibration algorithms, which do not require any specially designed calibration patterns or user interaction, simultaneously process several images captured by different cameras and find the correspondences across images. Correspondences are established by extracting 2D features from images automatically

and matching them between different images. Then, based on the established correspondences, camera locations and orientations can be estimated in a common coordinate system from the essential matrix.

A hierarchical method is proposed in [HL06]. It divides the set of cameras into several subgroups in which cameras share a common view. In each subgroup, three cameras' relative poses can be estimated by computing the associated trifocal tensor [HZ04] from point correspondences across the three views. After the triplets are registered into sub-groups, these subsets are then merged together to build the entire group. A coordinate system is established in each of these subgroups. In order to build a global coordinate system, different coordinate frames are merged in a hierarchical manner between the individual overlapping systems. In this hierarchical framework, the main advantage is that the error can be distributed evenly over the entire set of estimated camera matrices.

Rodehorst et al. [RHH08] proposed a self-calibration algorithm which can recover the relative pose between various image frames. These researchers compare various calibration techniques and analyze their difficulties on synthesis and real data. The drawbacks of relative orientation between two cameras are also presented. In order to overcome the drawbacks, the cameras are grouped in pairs with a fixed relation to each other. This implementation leads to additional constraints, which significantly stabilize the pose estimation process.

A fully automatic relative orientation estimation algorithm is presented in [LF06]. In this algorithm, the intrinsic parameters of cameras need to be calibrated first. Then, feature points on objects in the scene are extracted from all images and feature points in each image pair are matched using the SIFT parameters. No prior information about the overlapping areas among images is needed. The relative orientation between every image pair is derived using the RANSAC procedure and the 5-point algorithm [Nis04]. Based on the determined approximate orientations, camera orientations and object coordinates are computed. The best image pair is used to define the coordinate system. The other images are integrated sequentially,

increasing the bundle block step-by-step. Finally, the relative scale is determined by bundle adjustment and the camera poses are transformed into a common coordinate system.

Besides the algorithm which uses static features, Aslan et al. [ABS08] develop an algorithm which automatically calibrates camera networks using localized motion features. In contrast to the previously mentioned self-calibration approaches which extract feature points on static objects, this technique relies on the motion of persons walking naturally in the scene. Simple foreground and motion features are extracted from the individual image sequences. Using a Hough transform [DH72] in combination with a hierarchical gradient descent search method, the parameters of the pairwise camera geometries are then estimated, even in the presence of multiple moving objects. This procedure does not require the resolution of feature correspondences across camera views. After calibrating each camera pair, geometrical topology of the camera network is established using a global error minimization technique.

The accuracy of self-calibration is greatly dependent on the reliability of the relative pose estimates. This problem was first discussed in [DR04] with the concept of the *vision graph*. It models the set of uncalibrated cameras as nodes in a communication network, and a distributed algorithm is proposed, in which each camera communicates only with other cameras which have overlapping FoVs. Each node independently forms a neighborhood cluster on which the local calibration takes place, and calibrated nodes and scene points are incrementally merged into a common coordinate frame. Kurillo et al. [KLB08], Cheng et al. [CDR07], and Vergs-Llah et al. [VLMW08] later used and enhanced vision graph for this purpose. Vision graph G is used to represent a camera network consisting of M cameras. Graph G consists of M vertices ,V_i, which represents individual camera. In order to successfully achieve global calibration, the vision graph has to be connected, in terms of graph theory, which indicates that edges are established between some pairs (i, j) of vertices. If and only if two cameras have sufficient overlapping area

between their views, the vertices which represent these two cameras are connected. Weights are assigned to the graph edges. Weights can be determined by different error metrics, such as the number of matched feature points [CDR07, KLB08] and the unreliability measure of essential matrices [VLMW08]. After weights are assigned to edges, the calibration problem is transferred to the shortest path problem. We only need to find the shortest path between all vertices in the graph. The shortest path minimizes the estimation error in the calibration process. Vision graph is becoming a useful general tool for describing the directionality of networked visual sensors. *The invention of vision graph allows the calibration of camera setups in which all the cameras do not share a common working volume.* The only requirement is for the cameras to have pairwise overlapping FoVs. This approach has been more recently addressed by Bajramovic et al. [BD08, BBD12, BBD14]. They propose a graph-based calibration method which measures the uncertainty in the relative pose estimation between each camera pair.

All self-calibration algorithms take advantage of the epipolar structure of the system while suffering from scale ambiguity. This means that camera pose information can only be recovered up to a scale instead of the real world scale, because without a reference object with known dimensions in the scene, we cannot determine whether the camera is looking at a big object far away or a small object nearby.

2.3 Efficient Color and Depth Data Communication

In this section, we review the state-of-the-art algorithms for color and depth data communication. We first study the compression schemes for depth video. Next, we consider multi-view scenarios in which networked visual sensors have overlapping FoVs, and significant redundancy exists in the captured images. Finally, we review the image communication schemes which are especially designed for the multi-view scenarios.

2.3.1 3D Video Coding

Data compression of visual information is now a well-established technology. There are numerous lossless and lossy compression algorithms for image and video applications, JPEG 2000 [TM02] and H.264/AVC [WSBL03] being the most prominent. The coding of depth images, however, is a recent research topic. A depth image, representing the relative distance from the recording camera to an object in 3D space, usually consists of smooth regions and sharp edges at the boundaries between the object and the background. A typical way to compress a depth image sequence is by processing each frame as a standard gray scale image for viewing purposes and applying the standard video coding schemes. Redundancy between successive frames can be removed by estimating the motion between frames and then generating the motion vectors (MVs), which describe the motion of the pixel information repeatedly shown in successive frames. In standard video coding techniques, 2D block matching algorithms [PHS87] are used for motion estimation, in which frames are divided into blocks of $M \times N$ pixels, such as 16×16 and 8×8. A search is performed to find a matching block from a frame i in some other frame j. However, this approach is unfortunately very suboptimal for depth image sequences. Large, homogeneous areas on the surface of an object can be divided into small blocks and the sharp discontinuities at object boundaries can be placed into the same block. For these reasons, these schemes result in significant coding artifacts along the depth discontinuities in the reconstructed depth images, especially when the compression ratio is high [KCTS01, CSSH04].

The shortcomings of the standard image and video coding algorithms are the driving force for concentrating research efforts on developing new data compression schemes by considering the specific properties of depth images. The schemes can be broadly classified under two categories: Approaches developed to remove (i) temporal redundancies [GM04, HWDK08, DTPP09, SMAP14, KFMK09, FWL11, NMD13], or (ii) inter-view redundancies [ZHL10, SMW07, BAA06, EWK09, LWP11,

Feh04, LCH11, Mar99, WHY11] in depth video images. The starting point of all these schemes is 2D block matching algorithms. In the following paragraphs we provide a brief overview of these methods.

The method proposed in [GM04] exploits the correspondences between the depth images and the corresponding color frames captured by a texture camera. It adopts a conventional block matching approach to determine the MVs according to the texture information. The MVs from the texture information are considered to be encoding both the texture and depth image sequences. A number of techniques [DTPP09, KFMK09, FWL11, NMD13] have been developed based on this concept and provide enhanced performance. Daribo et al. [DTPP09] propose a MV sharing algorithm based on the correlation between the motion of the texture and of the depth. This algorithm considers the motion of a block at the same coordinates in both video texture and depth images and uses a joint distortion criterion to generate common MVs for both texture and depth. Shahriyar et al. [SMAP14] proposed an inherently edge-preserving depth-map coding scheme. Their scheme uses the texture motion vectors, avoids distortion on edges, and accurately preserves the depth information on edges. Fan et al. [FWL11] propose a motion estimation method with various block sizes. It first determines the block size and its corresponding MV using the color information. Then, a z-direction motion estimation is used to correct the depth values in each block. A similar algorithm, proposed in [KFMK09], replaces the 2D block matching algorithm with a 3D algorithm operating in horizontal, vertical, and depth dimensions. Hewage et al. [HWDK08] present a frame concealment method to deal with the case that a depth video frame is missing due to packet losses. In this case, the MVs of the correctly received corresponding color video frame are used as the candidate MVs for the current depth image frame to predict the missing depth frame at the enhancement layer decoder. The method of Nguyen et al. [NMD13] compresses the depth video by using a weighted mode filter to suppress the coding artifacts. The method by Oh et al. [OYVH09] focuses on post-processing and develops a depth reconstruction filter to recover the object

boundary and give advantages to both depth coding and rendering.

In addition to the temporal redundancy in the frames captured by a single camera at different times, inter-view redundancy also exists in the frames captured from various viewpoints in a multi-camera system. Many research studies have been proposed to remove these inter-view correlations and achieve efficient compression for depth video in multi-view scenarios. The study by Zhang et al. [ZHL10] combines multi-view video coding and the prediction of MVs in the depth image sequence from those of the texture image sequence. Ekmekçioğlu et al. [EWK09] propose a coding scheme in which the bit rate used for multi-view depth image coding is further reduced by skipping one or more temporal layers of selected depth image views. The method proposed by Lee et al. [LWP11] reduces the bit rate by skipping depth blocks based on consideration of temporal and inter-view correlations of texture images. The temporal and inter-view correlations are measured from the temporally successive pictures and the neighboring views synthesized by pixel-by-pixel mapping respectively. Lee et al. [LCH11] describe a multi-view depth video coding scheme that incorporates depth view synthesis, H.264/MVC, and additional prediction modes. A depth image-based rendering technique [Mar99] is adopted in this approach to generate an additional reference depth image for the current viewpoint. The inter-view correlation is exploited by using a texture and depth view synthesis approach. A block-based depth image interpolation approach is proposed by Wang et al. [WHY11]. In this scheme, the first and last frames in a texture video are treated as the key frames with known depth images. The remaining depth images corresponding to other texture frames can be recovered by the proposed bidirectional prediction algorithm.

To the best of our knowledge, these represent the most relevant works related to the depth video coding in which the characteristics of depth images are preserved and the temporal and inter-view correlations are exploited. However, all of these algorithms focus on compressing the depth images captured by cameras that remain in a fixed position. The situation becomes inevitably much more complicated when

moving cameras are involved. As the distance between a moving camera and the objects in a scene changes across time, depth values of the same objects change in successive depth frames. Therefore, coding schemes with motion compensation methods for static cameras become very inaccurate or even useless with mobile cameras.

2.3.2 Multi-View Image Compression and Transmission

Multiple visual sensors with overlapping FoVs provide multiple views, multiple resolutions and in that way, enhance observations of the environment, and become necessary in many applications such as object tracking, or 3-D reconstruction. In order to utilize the limited bandwidth efficiently in multi-view environments, many solutions have been proposed to exploit correlation and minimize the amount of redundant visual data transmitted. A number of studies in the research literature intend to remove or minimize correlated data for transmission in VSNs.

Since multi-view images are usually highly correlated, joint coding schemes [ML05, CAS09, CCAS12, CCM12, WC07] with encoders accessing images of multiple views achieve higher compression performance than traditional mechanisms with independent coding schemes. The spatial correlation can be explored and removed at encoders by image registration algorithms. Only the uncorrelated visual contents or low resolution images are delivered in the network after being jointly encoded by some latest coding techniques (e.g., Multiview Video Coding (MVC) [VWS11]).

A transmission framework consisting of three stages is presented in [ML05]. Camera sensors can cooperatively capture the scene and deliver partial information to the sink independently. In the first stage, camera sensors are grouped according to correlation degrees. Camera sensors with the maximum correlation are allocated to the same group. In order to obtain a fused image which is virtually taken at the center of the group, various sensing tasks are assigned to different sensors. In the second stage, camera sensors capture and deliver partial visual information

according to the result of the previous stage. In the final stage, once the images are received at the sink, they are fused together to construct a composite image.

Wagner et al. [WNB03] propose a collaborative in-network compression scenario with super-resolution recovery techniques applied at the receiver. In this scenario, in order to determine maximal overlap, images from correlated views are first registered using image matching. Then, the low-resolution version of the common image blocks describing the overlapping region is transmitted from each sensor to the receiver. Super-resolution techniques are applied at the receiver to reconstruct a high-resolution version of the overlapping region. In this study, the super-resolution algorithm requires a relatively large number of low-resolution images to reconstruct the overlapped region with an acceptable quality. Therefore, the camera sensors need to be deployed densely which effectively limits the flexibility and the coverage area of the network.

A collaborative image transmission system is developed in [WC07] which assumes that each sensor performs feature-based image matching locally and sensors on the route to the monitoring center can access image data collected from previous hops. At each sensor along the path, feature detection and matching operate on the image received from the previous hop (original image) and the image observed by current hop (reference image). In this phase, the redundancy between images is removed in a hop-by-hop manner. A transformation process operates to generate the difference image based on the result of redundancy removal. Then, only the original image and difference image are transmitted to the next hop.

Optimal fractions of the overlapped image are transmitted by various image sensors in the model proposed in [WPWS07]. The proposed model works in three phases. In the first phase, the overlapping regions and the non-overlapping regions in the images observed by multiple camera sensors are separated. In order to save energy at the sensors, each sensor transmits only visual data corresponding to non-overlapping region and a portion of the visual data corresponding to the overlapping region. Therefore, the correlated visual information of the overlapping

region can be avoided to be sent repeatedly. In the second phase, a multipath routing protocol is proposed to find the multiple node-disjoint routes from source sensors to the sink. In this protocol, sensors are classified into multiple levels according to their distance to the sink node, and lower data rates are assigned to sensor nodes with less residual energy. In the third phase, the optimized fractions of the visual information in overlapping regions are determined and transmitted by each sensor to maximize the network lifetime. Therefore, the total transmitted data is reduced throughout the network, and the end-to-end quality is preserved.

Chia et al. [CCAS12] use image stitching to exploit and remove the redundancy created by the overlapping FoVs. This system considers the memory requirement and the amount of computation for image stitching and performs image stitching and compression in a strip-by-strip manner. The stitching parameters are first determined after two reference images are transmitted to an intermediate node. These parameters are then sent back to the visual node. These parameters are used to determine the mechanism for stitching the incoming images in a strip-by-strip manner. After the stitching process is accomplished, the images can be further compressed using a strip-based compression technique.

A detailed discussion of multi-view image compression and transmission schemes in VSNs is presented in [WŞD13b]. However, in order to determine the overlapping regions in captured images, all of the above mentioned algorithms require at least one node in the network to have full knowledge of images captured by the other sensors. This indicates that the redundant information cannot be removed completely and still needs to be transmitted at least one time. Moreover, as color images do not contain full 3D representation of a scene, these methods introduce distortions and errors when the relative poses between sensors are not pure rotation or transition, and the scenes have complex geometrical structures and occlusions.

The algorithms mentioned above focus only on color information with no exception. Only a few studies have been reported [AJ13, SSCZ13] which use RGB-D

sensors in VSNs as their use in VSNs has not yet become ubiquitous. Our extensive review of the research literature has identified that no earlier studies have been published that attempt to develop an efficient coding system considering both color and depth information for optimizing the bandwidth usage for wireless communications in VSNs. Some multi-view depth video coding schemes introduced in Section 2.3.1 can remove the redundancy in multi-view situations, however the processing nodes in these schemes always have the full knowledge of the images captured by all cameras. This characteristic is different from VSNs, as in VSNs each visual sensor only has its own captured images. Therefore, these algorithms cannot be applied to solving redundancy removal problems in VSNs.

2.4 Summary

In this chapter, we comprehensively review the current research on four topics: (1) depth image registration for RGB-D sensor pose estimation, (2) self-calibration methods, (3) 3D video coding, and (4) multi-view image communication. The first two topics focus on the pose estimation of visual sensors, while the latter two topics are related to achieving efficient color and depth information communication. Due to the differences between RGB-D sensors and conventional cameras, there are still many virgin areas in pose estimation and efficient communication problems for RGB-D camera-equipped VSNs.

Depth image registration for sensor pose estimation

Although many methods have been proposed to determine a single mobile RGB-D sensor's motion by matching the consecutive depth frames, no studies have been reported on relative pose estimation between multiple RGB-D sensors. There are two main differences between the motion estimation of a single sensor and the relative pose estimation of multiple sensors: (1) as each sensor only has the information on its captured images, the motion estimation algorithms can be

centralized while the relative pose estimation algorithms have to be distributed, (2) the relative pose between two sensors is usually much larger than the inter-frame motion of a single mobile sensor, which makes the depth images hard to register. These two major differences prevent the existing algorithms from being used to estimate relative pose directly.

Self-calibration

There have been few studies on the calibration of a VSNs equipped with RGB-D cameras. Although many self-calibration algorithms have been proposed for RGB cameras, these algorithms, based on epipolar geometry, suffer from scale ambiguity. As the RGB-D sensor provides range information along with the color images, the locations and orientations of RGB-D sensors can be determined on the real world scale without involving scale ambiguity problem.

3D video coding

Existing 3D video coding schemes focus on compressing the depth images captured by cameras that remain in a fixed position. These algorithms are all based on the assumption that the differences of two successive depth frames are mostly zero, except in the motion area where variance of depth can be observed. This statement is no longer correct when the camera is moving. Because the distance between a moving camera and the objects in a scene changes across time when the camera is moving, even the static areas in the scene can have various depth information between two successive frames. Therefore, coding schemes with motion compensation methods for static cameras become very inaccurate or even useless with mobile cameras.

Multi-view image communication

Although many methods have been proposed to compress multi-view images, they cannot be applied in our circumstances, because these approaches either require

the transmitter to have knowledge of the full set of images or only work on cameras with very small pose differences. In our case, each sensor only has it own captured images and the pose difference between two visual sensors can be large. Therefore, we need to develop a new algorithm to achieve efficient color and depth data communication for VSNs equipped with RGB-D cameras.

In the following chapters, we propose new methods to fill the above mentioned gaps in these four topics.

CHAPTER THREE

RELATIVE POSE ESTIMATION BETWEEN TWO RGB-D SENSORS

In this chapter, we describe a method for determining the relative pose between two RGB-D sensors in indoor environments. Each RGB-D sensor is a robot with an RGB-D camera mounted on the top, named "eyeBug". Our algorithm is based on the ICP algorithm, but explicitly accounts for the situation where two views of a scene each see parts that are occluded in the other view by making use of a beam imaging model implemented by reweighting the least squares operation in ICP. Further, we show how by symmetrizing across the two views, the bias that beam models introduce can be eliminated. Finally, our algorithm makes sensing errors isotropic by operating in inverse depth coordinates. Sections of this chapter have been published in a conference paper [WŞD13a].

3.1 Introduction

The latest advances in depth-sensing technology enable the wide utilization of inexpensive RGB-D cameras which can capture color images along with per-pixel depth information. RGB-D camera-equipped VSNs, by using the additional depth data, can significantly enhance the performance of conventional collaborative tasks such as immersive telepresence and mapping [SEE+12, BSK+13, TZL+12], environment surveillance [CPS11, LXW+12], object recognition and tracking [BCB+12, MMN13, AJ13], and present possibilities for new and innovative applications

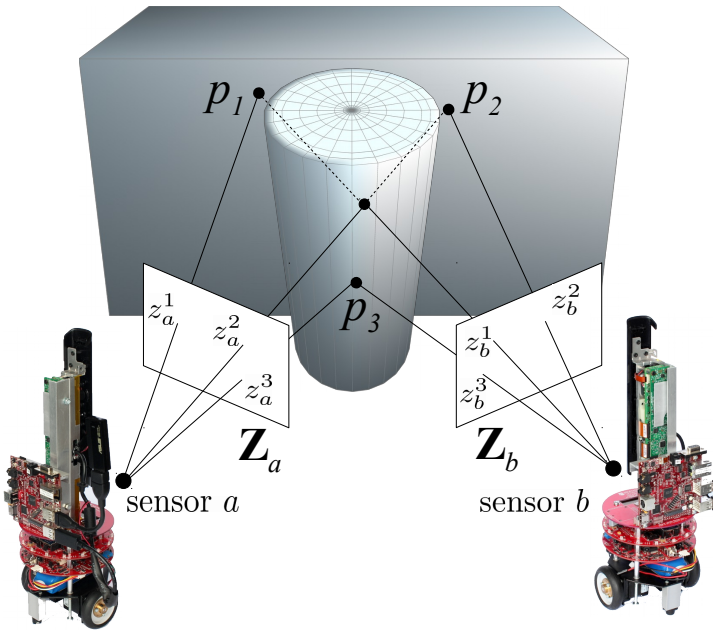


Figure 3.1: A scene with occlusion: $\mathbf{Z}_a$ and $\mathbf{Z}_b$ denote a pair of depth images. p_3 is a world point which can be seen by both Sensors a and b. p_1 and p_2 are world points which can only be seen by either Sensor a or Sensor b.

[HSXS13, HKH+12].

Collaborative tasks between multiple sensors in VSNs critically rely on accurate extrinsic calibration, i.e., the knowledge of the cameras' relative pose. In this chapter, we address the problem of determining the 6 DoF relative pose of two RGB-D sensors in indoor environments. Each sensor is comprised of a RGB-D camera and a robot platform with local processing ability. The goal is to enable each RGB-D sensor to obtain the precise location and orientation information of other sensors.

A popular algorithm for achieving this is the ICP algorithm, both in point-to-point and point-to-plane variants. This algorithm samples surface points from one view and attempts to compute the transformation that places them onto the surface in the other view. At each iteration it attempts to find the closest point on the surface to each sample point and to find an update to the transformation that minimizes the sum squared distance between each sample point and the surface. Through

matching these point clouds, the transformation between them and, consequently, the 6 DoF relative pose between two RGB-D cameras can be deduced.

One complexity that arises in this approach is that the sample point may be occluded in the view from which the surface was seen and thus there may be a discrepancy between the sample point and the surface (e.g. Figure 3.1). This is often handled by including an outlier check that excludes points that are too far from the expected surface [PLT07]. This approach can be improved by noting that the sample point from Sensor a's view can only lie on or behind the surface as seen from Sensor b's view. If it lies in front of the surface then Sensor b should have seen it. Therefore, points on or in front of the surface should play a more important role in the minimization than points behind it. This gives rise to the beam model [TBF05] which explicitly encodes this asymmetry in a model of the probability distribution for signed distance errors between the sample point and the surface.

This model creates a bias, however. It tends to push the sample points away from the camera that sensed the surface so that they are more likely to lie behind it. In this chapter, we propose an algorithm which performs ICP using a beam model bidirectionally, i.e. by sampling points from both surfaces and computing a single transformation between the views that causes the sample points to lie on or behind the surface as seen from the other view. This approach removes the bias that the beam model introduces, and gives more accurate and robust results. We implement our beam model by using an asymmetric weight function in the least squares component of our ICP solver. Then, in order to make the algorithm more practical, we distribute the working load to two RGB-D sensors and implement the algorithm on a real VSN.

The main contributions of this chapter are:

- A novel bidirectional beam-based sensor model which uses probability to identify and deal with occlusion. By making the beam-based sensor model work in a bidirectional way, the proposed model can handle occlusion more reasonably and prevent point cloud alignment from being ill-posed.

- A theoretical framework and practical implementation of the ICP algorithm with the bidirectional beam-based model. Inverse depth coordinates are adopted to reduce uncertainty of depth measurements, non-homogeneous error and anisotropic noise.
- The algorithm was implemented and tested both on a laptop and our VSN testbed of mobile RGBD sensors.
- Extensive experiments using real world data were conducted to evaluate the performance of proposed algorithm.

3.2 Problem Statement

As an RGB-D sensor can provide a continuous measurement of the 3D structure within the environment, the relative pose between two RGB-D sensors can be estimated through explicit matching of surface geometry. The relative pose between two sensors a, b can be represented by a transformation matrix, $\mathbf{M}_{ab}$, in SE(3),

$$\mathbf{M}_{ab} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{R}$ is a 3×3 matrix indicating the relative orientation, and $\mathbf{t}$ is a 3×1 vector representing the relative position. The subscript ab indicates Sensor a's pose relative to Sensor b's pose.

Let $\mathbf{Z}_a$ and $\mathbf{Z}_b$ denote a pair of depth images of the same scene captured by two separated RGB-D sensors (see Fig. 3.1). The depth pixels in the depth image $\mathbf{Z}_a$ can be mapped to $\mathbf{Z}_b$. Consider the vector $\mathbf{p}_e$ which represents a real world point in Euclidean space. In the discussion that follows, we assume that $\mathbf{p_e}$ can be observed in $\mathbf{Z}_a$ and $\mathbf{Z}_b$ captured by two RGB-D sensors a and b. Therefore, real point $\mathbf{p}_e$ can be expressed in Sensor a and Sensor b's coordinate systems by using homogeneous

coordinates respectively,

$$\mathbf{p}_e = [x_a \; y_a \; z_a \; 1]^T, \tag{3.2}$$

$$\mathbf{p}_e = [x_b \; y_b \; z_b \; 1]^T. \tag{3.3}$$

The projections of $\mathbf{p}_e$ are located at pixel coordinates (i_a, j_a) and (i_b, j_b) on the depth images $\mathbf{Z}_a$ and $\mathbf{Z}_b$, respectively. Given the intrinsic parameters of the RGB-D sensor a: principal point coordinates $(i_{c,a}, j_{c,a})$ and focal length of the camera $(f_{x,a}, f_{y,a})$, $\mathbf{p}_e$ can be estimated from the corresponding pixel in depth image $\mathbf{Z}_a$ by using the pinhole camera model as

$$\mathbf{p}_e \equiv \frac{1}{z_a}\begin{bmatrix} x_a & y_a & z_a & 1 \end{bmatrix}^T \equiv \begin{bmatrix} \frac{i_a - i_{c,a}}{f_{x,a}} & \frac{j_a - j_{c,a}}{f_{y,a}} & 1 & \frac{1}{z_a} \end{bmatrix}^T = \begin{bmatrix} u_a, v_a, 1, q_a \end{bmatrix}^T, \tag{3.4}$$

where (i_a, j_a) denotes the pixel coordinates of this real world point projection in the depth image $\mathbf{Z}_a$, and z is the corresponding depth value reported by Sensor a. Similarly, the relation between $\mathbf{p}_e$ and its projection on image $\mathbf{Z}_b$ can be expressed as,

$$\mathbf{p}_e \equiv \begin{bmatrix} \frac{i_b - i_{c,b}}{f_{x,b}} & \frac{j_b - j_{c,b}}{f_{y,b}} & 1 & \frac{1}{z_b} \end{bmatrix}^T = \begin{bmatrix} u_b, v_b, 1, q_b \end{bmatrix}^T, \tag{3.5}$$

It is more convenient to solve for pose using this format, because (u, v) are a linear function of pixel position. It preserves the linear relationship with the normalized disparity values and avoids conversion to 3D Euclidean space which has non-homogeneous and anisotropic noise characteristics. With the accurate information of the transformation matrix, the depth pixel (projection) at (i_a, j_a) in $\mathbf{Z}_a$ can establish a relationship between the depth pixel at (i_b, j_b) in $\mathbf{Z}_b$ as follows,

$$\begin{bmatrix} \frac{i_b - i_{c,b}}{f_{x,b}} & \frac{j_b - j_{c,b}}{f_{y,b}} & 1 & \frac{1}{z_b} \end{bmatrix}^T = \mathbf{M}_{ab} \begin{bmatrix} \frac{i_a - i_{c,a}}{f_{x,a}} & \frac{j_a - j_{c,a}}{f_{y,a}} & 1 & \frac{1}{z_a} \end{bmatrix}^T \tag{3.6}$$

and, to simplify the equation, by doing some rudimentary algebraic substitutions

we obtain the following equation in inverse depth coordinate,

$$[u_b \quad v_b \quad 1 \quad q_b]^T = \mathbf{M}_{ab} [u_a \quad v_a \quad 1 \quad q_a]^T . \tag{3.7}$$

Conversely, if we can establish the correspondences between two depth images and put the corresponding pixel pairs in Eq. 3.6, the transformation matrix denoting the relative pose between two RGB-D sensors can be determined. Moreover, all the pixels in $\mathbf{Z}_a$ can be warped to generate a virtual depth image which matches $\mathbf{Z}_b$.

However, when there is occlusion in the scene (see Fig. 3.1), some world points may only be seen by Sensor a and cannot be seen by Sensor b. Therefore, the pixels representing these points in $\mathbf{Z}_a$ are not able to find their correct corresponding pixels in $\mathbf{Z}_b$. If the incorrect correspondences are established, a virtual depth image which cannot match $\mathbf{Z}_b$ will be generated. Therefore, a wrong transformation matrix is provided according to Eq. 3.6.

3.3 Sensor Model in a Maximum Likelihood Framework

In the research literature, beam models have been applied to the motion estimation problem using a maximum likelihood framework [KKF12]. In this work [KKF12], the beam model was applied unidirectionally by taking one of the two images to be aligned as the reference. This introduces a bias into the model. To remove this bias, we propose to use the beam model bidirectionally. In this section, we first review how the unidirectional beam model is used for motion estimation. Then, we describe how the bidirectional beam model can be formulated within a maximum likelihood framework. We show that the bidirectional beam model in this form is difficult to resolve and costly to compute. This forms the basis for our motivation to incorporate the beam model as a robust weighting function in ICP, as explained in Section 3.4.

3.3.1 Beam-based Sensor Model

Let $\mathcal{D}_a$ and $\mathcal{D}_b$ denote the depth measurements returned by the Sensor a and b. Each set of the depth measurements is made up of N pixel elements where each pixel in the image contains the corresponding depth value, z_a^k, such that $\mathcal{D}_a = \{z_a^1, ..., z_a^N\}$. In this model, the depth information in $\mathcal{D}_b$ is treated as the expected surface, and the depth information in $\mathcal{D}_a$ is treated as the measurements. The relative motion which best aligns the measurements to the expected surface, described by a 6DoF motion matrix $\mathbf{M}_{ab}$, can be estimated by formulating this as a maximum likelihood problem as follows,

$$\mathbf{M}_{ab} = \underset{\widetilde{\mathbf{M}}_{ab}}{\arg\max}\, p(\mathcal{D}_a|\mathcal{D}_b, \widetilde{\mathbf{M}}_{ab}). \tag{3.8}$$

The conditional probability $p(\mathcal{D}_a|\mathcal{D}_b, \widetilde{\mathbf{M}})$ can be approximated by the product of the individual measurement probabilities:

$$p(\mathcal{D}_a|\mathcal{D}_b, \widetilde{\mathbf{M}}_{ab}) = \prod_k p(z_a^k|\mathcal{D}_b, \widetilde{\mathbf{M}}_{ab}), \tag{3.9}$$

where $p(z_a^k|\mathcal{D}_b, \widetilde{\mathbf{M}}_{ab})$ can be modeled according to the beam model which describes the probability distribution of a measurement z_a^k lying in front (occluding surface), close to, or beyond the surface given the expected measurements in the depth measurements $\mathcal{D}_b$ and the motion matrix $\mathbf{M}_{ab}$. The beam model is illustrated in Fig. 3.2 and this can be represented using a piecewise function. There are three parts in this piecewise function:

- Case 1: when $z_a^k \ll z_b^k$: describes the probability of a depth measurement z_a^k being an occluded surface. This is described by a uniformly distributed function.
- Case 2: when $z_a^k \approx z_b^k$: describes the probability of a depth measurement z_a^k

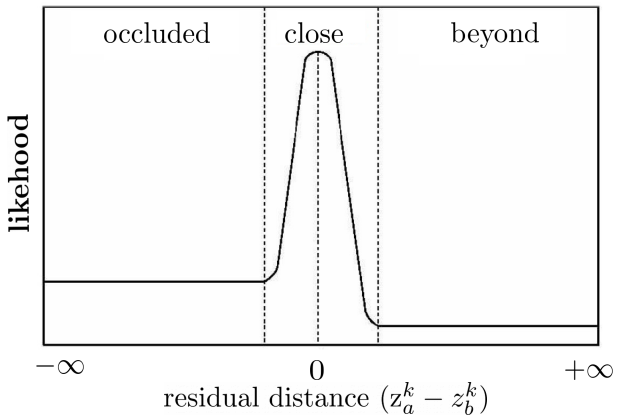


Figure 3.2: Piecewise function used in the beam model.

being closely aligned to its expected value. This is described by a Gaussian distribution centered at 0 with standard deviation σ_z for $z_a^k \approx z_b^k$

- Case 3: when $z_a^k \gg z_b^k$: describes the probability of a depth measurement z_a^k lying beyond its expected value. This is described by a uniformly distributed function with very low probability.

The beam model when applied to motion estimation within a maximum likelihood framework explicitly deals with occlusion. However, as the piecewise model used to distinguish various points is asymmetrical, the beam model creates a bias. It tends to push the sample points away from the camera that senses the surface so that they are more likely to lie behind it, which would produce less robust and more inaccurate results.

3.3.2 Bidirectional Beam Model

To remove this bias introduced by beam model, we propose to use the beam model bidirectionally, and we name it the bidirectional beam model. In a maximum likelihood framework, this can be formulated as

$$\mathbf{M}_{ab} = \arg\max_{\widetilde{\mathbf{M}}_{ab}} [p(\mathbf{Z}_a|\mathbf{Z}_b, \widetilde{\mathbf{M}}_{ab}) p(\mathbf{Z}_b|\mathbf{Z}_a, \widetilde{\mathbf{M}}_{ab}^{-1})]. \tag{3.10}$$

where $\mathbf{Z}_a$, $\mathbf{Z}_b$ are the depth images captured by Sensor a and Sensor b respectively. Similar to the unidirectional beam model, we can assume the conditional independence for each depth measurement such that

$$p(\mathbf{Z}_a|\mathbf{Z}_b, \widetilde{\mathbf{M}}) = \prod_k p(z_a^k|\mathbf{Z}_b, \widetilde{\mathbf{M}}_{ab}), \tag{3.11}$$

$$p(\mathbf{Z}_b|\mathbf{Z}_a, \widetilde{\mathbf{M}}_{ab}^{-1}) = \prod_k p(z_b^k|\mathbf{Z}_a, \widetilde{\mathbf{M}}_{ab}^{-1}). \tag{3.12}$$

According to the piecewise function of the beam model, an occlusion determined by $p(\mathbf{Z}_a|\mathbf{Z}_b, \widetilde{\mathbf{M}}_{ab})$ is treated as the measurement beyond the expected surface by $p(\mathbf{Z}_b|\mathbf{Z}_a, \widetilde{\mathbf{M}}_{ab}^{-1})$. When the first probability component is maximized as points are being pushed to the front of the reference surface, the second probability component will become smaller, which can prevent the transformation matrix $\mathbf{M}_{ab}$ being incorrectly estimated. Eq. 3.10 can only be maximized when the balance between the two probability components is reached. Eq. 3.10 can be converted into negative log likelihoods,

$$\mathbf{M}_{ab} = \underset{\widetilde{\mathbf{M}}_{ab}}{\arg\min} \sum_k [\log p(z_a^k|\mathbf{Z}_b, \widetilde{\mathbf{M}}_{ab}) + \log p(z_b^k|\mathbf{Z}_a, \widetilde{\mathbf{M}}_{ab}^{-1})]. \tag{3.13}$$

To estimate the 6DoF motion that best aligns a pair of depth images in this form would require partial derivatives of $\widetilde{\mathbf{M}}_{ab}$ with respect to the 6 motion parameters. However, as Eq. 3.13 contains both $\mathbf{M}_{ab}$ and $\mathbf{M}_{ab}^{-1}$, we face the difficult high dimensional partial differential problem. This is not a trivial task, and even if this is achievable, it is not computationally efficient. This forms the basis of our motivation to incorporate the bidirectional beam model as a robust weighting function in the ICP algorithm.

The advantages of using the bidirectional beam model for pose estimation include:

- It clearly classifies sensed range data into three categories: occluded, near and beyond.
- By achieving balance between alignments in two directions, it eliminates the alignment bias of the beam-based sensor models, while preserving their original nice properties.

3.4 Motion Estimation Using ICP with Bidirectional Beam Model

In this section, we describe how we incorporate the bidirectional beam model into the ICP algorithm for 6DoF pose estimation. We first describe various stages in the standard ICP algorithm and illustrate a variant of ICP using the point-to-plane metric. Then, we describe how the bidirectional beam model is incorporated into the ICP algorithm using an asymmetric weight function in the least squares component of our ICP solver.

3.4.1 ICP Algorithm

The standard ICP algorithm can be described by the following stages,

1. Selection - select N number of points from a reference depth image;
2. Matching - establish corresponding points for selected points in corresponding depth image;
3. Weighting (optional) - Weigh the correspondences based on some measure of confidence that indicates the quality of the correspondences;
4. Rejection (optional) - Reject point correspondences using fixed thresholds;
5. Error metric - Minimize error metric and estimate the motion vector α_j described in Section 4.1. This is then used to update the motion matrix M;
6. Iterate from the first stage until convergence.

There are two popular error metrics in the ICP algorithm – point-point error metric and point-to-plane error metric. As the point-to-plane error metric is in general superior to the point-to-point error metric [PCSM13], we adopt ICP with the point-to-plane error metric in this thesis. The point-to-plane error metric expressed in normal least squares form is,

$$\mathcal{C} = \sum_{l=1}^{N} \left[w_{l,a} (\mathbf{M}_{ab} \mathbf{p}_a^l - \mathbf{p}_b^{l*}) \cdot \vec{n}_{l*,b} \right]^2, \tag{3.14}$$

where $\mathbf{p}_a^l$ are the sampled points in depth frames $\mathbf{Z}_a$, and $\mathbf{p}_b^{l*}$ are their corresponding points on $\mathbf{Z}_b$. The variables $w_{l,a}$ are weight parameters for established correspondences, and $\vec{n}_{l*,b}$ are the surface normals at the corresponding points $\mathbf{p}_b^{l*}$ in real world coordinates. This cost function indicates the error between the established correspondences between the depth images captured by two sensors. Transformation matrix, $\mathbf{M}_{ab}$, can be determined by iteratively minimizing Equation (3.14).

3.4.2 ICP with Bidirectional Beam Model

We now describe how the bidirectional beam model is incorporated into the ICP algorithm. We approach this problem by using an asymmetric weighting function in the least squares component of our ICP solver. As reported in [HW77], different weighting functions lead to various probability distributions. For a weighting function $w(x)$, the probability density function is expressed as,

$$p(x) = \frac{1}{k} \exp\left(-\lambda \int_0^x x' w(x') \mathrm{d}x' \right), \tag{3.15}$$

where $k = \int_{-\infty}^{+\infty} \exp\left(-\lambda \int_0^x x' w(x') \mathrm{d}x' \right)$ is the normalization factor. To achieve the probabilistic model for the beam model in Fig. 3.2, we find a piecewise weighting

function as follows,

$$w(z) = \begin{cases} c/[c + (z^* - z)] & \text{if } z \leqslant z^* \\ c/[c + (z^* - z)^2] & \text{if } z > z^* \end{cases}, \tag{3.16}$$

where z^* is the expected depth value, and z is the measured value. c is the mean of deviation between expected and measured depth values.

As shown in Fig. 3.2, the likelihood of one correspondence is directly related to the residual distance between the measurement and the expected surface. Therefore, the maximum likelihood framework of the bidirectional beam model can be converted and solved as a novel least squares approach which operates in a bidirectional way with the weighting function presented above. If correspondences between $N = N_a + N_b$ pairs of points from two depth images $\mathbf{Z}_a$ and $\mathbf{Z}_b$ are established, we can then estimate the transformation matrix $\mathbf{M}_{ab}$ by minimizing

$$\begin{aligned} \mathcal{C} = & \sum_{l=1}^{N_a} \left[w_{l,a} (\mathbf{M}_{ab} \mathbf{p}_a^l - \mathbf{p}_b^{l^*}) \cdot \vec{n}_{l^*,b} \right]^2 + \\ & \sum_{k=1}^{N_b} \left[w_{k,b} ({\mathbf{M}_{ab}}^{-1} \mathbf{p}_a^{k^*} - \mathbf{p}_b^k) \cdot \vec{n}_{k,b} \right]^2, \end{aligned} \tag{3.17}$$

where $\mathbf{p}_a^l$ and $\mathbf{p}_b^k$ are the sampled points in depth frames $\mathbf{Z}_a$ and $\mathbf{Z}_b$, $\mathbf{p}_b^{l^*}$ and $\mathbf{p}_a^{k^*}$ are their corresponding points on the other depth image respectively. The variables $w_{l,a}$ and $w_{k,b}$ are weight parameters for correspondences established in opposite directions between pairs. The variables $w_{l,a}$ and $w_{k,b}$ are weight parameters for correspondences established in opposite directions between pairs. In addition, $\vec{n}_{l^*,b}$ and $\vec{n}_{k,b}$ are the surface normals at the corresponding points $\mathbf{p}_b^{l^*}$ and $\mathbf{p}_b^k$ in real world coordinates, and

$$\vec{n}_{l^*,b} = \begin{bmatrix} \alpha_{l^*,b} & \beta_{l^*,b} & \gamma_{l^*,b} & 0 \end{bmatrix}^T, \tag{3.18}$$

$$\vec{n}_{k,b} = \begin{bmatrix} \alpha_{k,b} & \beta_{k,b} & \gamma_{k,b} & 0 \end{bmatrix}^T. \tag{3.19}$$

The cost function presented in Eq. 3.17 consists of two parts:

1. the sum of squared distances in the forward direction from depth images $\mathbf{Z}_a$ to $\mathbf{Z}_b$, and

2. the sum of squared distances in the backward direction from $\mathbf{Z}_b$ to $\mathbf{Z}_a$.

Algorithm 1 Relative pose estimation procedure

1: Capture a depth image, $\mathbf{Z}_a$, on robot a, and capture a depth image, $\mathbf{Z}_b$, on robot b.
2: Initialize the transformation matrix, $\mathbf{M}_{ab}$, by the identity transformation.
3: **procedure** REPEAT UNTIL CONVERGENCE
4: Update depth frame $\mathbf{Z}_a$ according to transformation matrix.
5: Randomly sample N_a points from $\mathbf{Z}_a$ to form set P_a,
$S_a = \{\mathbf{p}_a^k \in \mathbf{Z}_a, k = 1, \ldots, N_a\}$,
6: Randomly sample N_b points from $\mathbf{Z}_b$ to form set P_b,
$S_b = \{\mathbf{p}_b^k \in \mathbf{Z}_b, k = 1, \ldots, N_b\}$.
7: Find the corresponding point set, P_b^*, of P_a in $\mathbf{Z}_b$,
$S_b^* = \{\mathbf{p}_b^{k*} \in \mathbf{Z}_b, k = 1, \ldots, N_a\}$;
Find the corresponding point set, P_a^*, of P_b in $\mathbf{Z}_a$,
$S_a^* = \{\mathbf{p}_a^{k*} \in \mathbf{Z}_a, k = 1, \ldots, N_b\}$.
▷ The correspondences are established using the project and walk method with a neighborhood size of 3x3 based on the nearest neighbor criteria
8: Apply the weight function bidirectionally,
$S_a \mapsto S_b^*$, $S_b \mapsto S_a^*$
9: Compute and update transformation matrix based on current bidirectionally weighted correspondences
10: **end procedure**

An overview of the entire process is presented in Algorithm 1. In this coarse-to-fine algorithm, each iteration generates an update $\mathbf{E}$ to the sensor's pose which modifies the transformation matrix $\mathbf{M}_{ab}$. $\mathbf{E}$ takes the same form as $\mathbf{M}_{ab}$, which may be parameterized by a 6-dimensional motion vector having the elements $\alpha_1, \alpha_2, \ldots, \alpha_6$ via the exponential map and their corresponding group generator matrices $\mathbf{G}_1, \mathbf{G}_1, \ldots, \mathbf{G}_6$ as

$$\mathbf{E} = \exp\left(\sum_{j=1}^{6} \alpha_j \mathbf{G}_j\right), \tag{3.20}$$

where

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{G}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G}_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{G}_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here $\mathbf{G}_1$, $\mathbf{G}_2$ and $\mathbf{G}_3$ are the generators of translations in x, y and z directions, while $\mathbf{G}_4$, $\mathbf{G}_5$ and $\mathbf{G}_6$ are rotations about x, y and z axes respectively.

The task then becomes finding the $\alpha_1, \ldots, \alpha_6$ that describe the relative pose. Through determining the partial derivatives of u_b, v_b and q_b with respect to the unknown elements of the motion vector $\alpha_1, \ldots, \alpha_6$, the Jacobian matrix for each established corresponding point pair can be obtained from

$$\mathbf{J} = \begin{bmatrix} q_a & 0 & -u_a q_a & -u_a v_a & 1 + u_a^2 & -v_a \\ 0 & q_a & -v_a q_a & -1 - v_a^2 & v_a u_a & u_a \\ 0 & 0 & -q_a^2 & -v_a q_a & u_a q_a & 0 \end{bmatrix}. \tag{3.21}$$

The six-dimensional motion vector, which minimizes Eq. 3.17, is then determined iteratively by the least squares solution

$$\mathbf{B} = (\mathbf{K}^T \mathbf{W} \mathbf{K})^{-1} \mathbf{K}^T \mathbf{W} \mathbf{Y} \tag{3.22}$$

in which $\mathbf{W}$ is a diagonal matrix weighting the bidirectional point-to-plane corre-

spondences, and **B**, **Y**, and **K** are matrices,

$$\mathbf{B} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} -(\mathbf{p}_a^1 - \mathbf{p}_b^{1*}) \cdot \vec{n}_{1*,b} \\ \vdots \\ -(\mathbf{p}_a^{N_a} - \mathbf{p}_b^{N_a^*}) \cdot \vec{n}_{N_a^*,b} \\ -(\mathbf{p}_a^{1*} - \mathbf{p}_b^1) \cdot \vec{n}_{1,b} \\ \vdots \\ -(\mathbf{p}_a^{N_b^*} - \mathbf{p}_b^{N_k}) \cdot \vec{n}_{N_b,b} \end{bmatrix}, \tag{3.23}$$

$$\mathbf{K} = \begin{bmatrix} \vec{n}'_{1*,b}\mathbf{J}_1 \ ... \ \vec{n}'_{N_a^*,b}\mathbf{J}_{N_a} \ \vec{n}'_{1,b}\mathbf{J}_{1*} \ ... \ \vec{n}'_{N_b,b}\mathbf{J}_{N_b^*} \end{bmatrix}^T. \tag{3.24}$$

Here, $\vec{n}'_{l,b} = \begin{bmatrix} \alpha_{l,b} & \beta_{l,b} & \gamma_{l,b} \end{bmatrix}^T$ is the surface normal expressed in a slightly different form than that shown in Eqs. 3.18 and 3.19. To detect the convergence of our algorithm, we use the thresholds for the ICP framework presented in [LTDL12]. Once the algorithm converges, the registration is considered completed, and the $\mathbf{M}_{ab}$ is refined based on the initial relative pose.

3.4.3 Distributing the Algorithm to Two RGB-D Sensors

In reality, each sensor has only its own captured depth frames. In order to accomplish the central working principle of the algorithm described above, we distribute the algorithm to two sensors.

Considering the limited bandwidth of the network, instead of transmitting a complete depth image from one sensor to another, each sensor transmits only a number of sampled points to the other sensor. For example, at each iteration, after Sensor b receives the sampled point set, P_a, from Sensor a, Sensor b will find the corresponding point set, P_b^*, on its captured depth frame $\mathbf{Z}_b$. The first component in Eq. 3.17 will be derived. The information representing the first component will be sent with the sampled point set, P_b, from Sensor b to sensor a. At Sensor a, P_b's corresponding point set, P_a^*, will be determined. The second component in Eq. 3.17

will be derived. Thereby, Sensor *a* will acquire the information of both the first and second components in Eq. 3.17. The motion parameters can be determined.

These procedures are performed in each iteration. The transformation matrix describing the relative pose between two sensors is obtained by Sensor *a* until the algorithm converges. Sensor *a* sends the inverse transformation matrix to sensor *b*. Then, both sensors obtain the information on the other sensor's location and orientation. The distributed process is illustrated in Fig. 3.3.

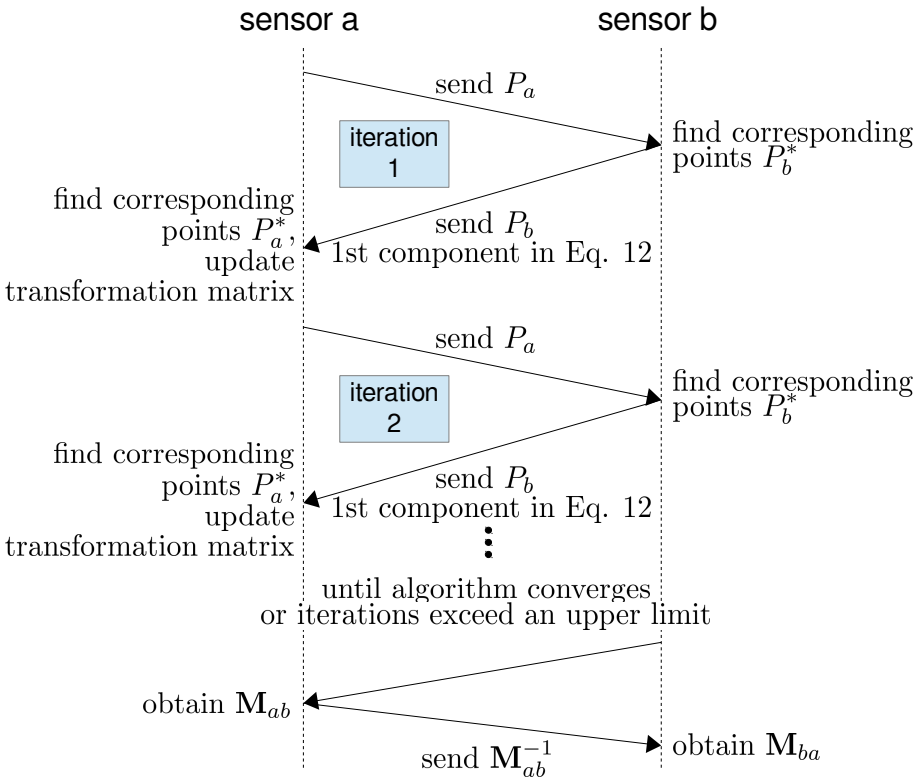


Figure 3.3: Distributing the tasks to two mobile sensors.

3.5 Experimental Results and Discussion

In order to justify the proposed algorithm for relative pose estimation between multiple RGB-D sensors, we conducted extensive tests to evaluate the performance. We implemented our algorithm (ICP-BD) in C++ using the libCVD [lib] and OpenKinect [opeb] libraries on a laptop with an Intel i7 M620 processor and our mobile VSN testbed to evaluate its fast processing and robust performance. To verify the

superiority of our algorithm in relative pose estimation, we compared it with the benchmark ICP algorithm [BM92] and ICP in inverse depth coordinates (ICP-IVD) [LTDL12] using point-to-plane error metric.

3.5.1 Dataset Simulations

This set of experiments was conducted on a laptop using the datasets *Cabinet, Large cabinet, Plant, and Structure-no-texture* provided in [SEE+12]. Each dataset is a sequence of Kinect video frames capturing one scene from different angles of view. In order to simulate situations including different amounts of occlusion between two sensors' views, we extracted 4 new sequences from each dataset by taking one frame out of every 5, 10, 20, and 30 frames. For each trial we treated two consecutive frames in the new sequence as the depth images captured by two separated sensors. We deemed a trial to be successful if the error between the estimated pose and ground truth pose was within 10 centimeters in translation. The percentages of successful relative pose estimation for different algorithms are presented in Fig. 3.4.

Fig. 3.4 clearly indicates that

- As the frame is sampled at an incremental interval, each algorithm's successful percentage decreases. When the frame interval is greater than 10, more occlusions and differences exist between two sensors' views. As the proposed ICP-BD reports higher successful estimation percentages, it outperforms other algorithms in environments with heavy occlusion.
- When the frame interval is 5, the three algorithms have similar performances. Therefore, all can be used to handle small motion in the presence with minimal occlusion.
- When the frame interval is 30, the occlusion between two views is too heavy and the two consecutive depth images the different from each other. As a result, the performance of three algorithms reduces significantly.

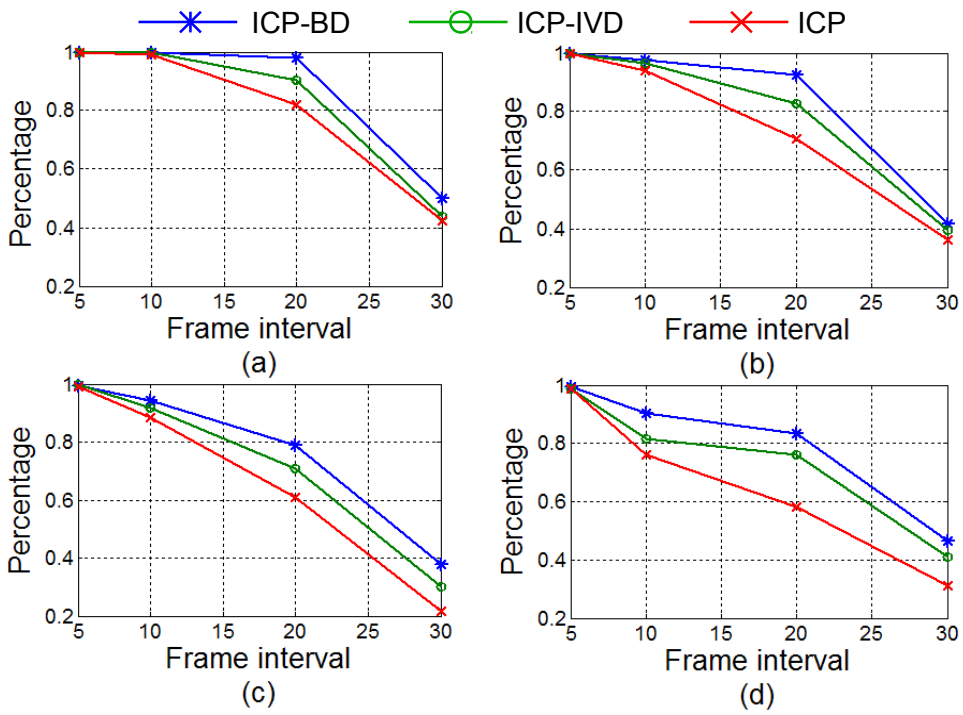


Figure 3.4: Percentage of successful relative pose estimation in various datasets: (a) cabinet, (b) large cabinet, (c) dish, and (d) plant by three algorithms: ICP-BD, ICP-IVD, and ICP.

Furthermore, by adjusting the number of sample points on the depth frames, our proposed algorithm can process up to 30Hz while still maintaining estimation accuracy on a standard laptop without GPU implementation, which is faster than the other ICP variants.

3.5.2 Turntable Simulations

In order to precisely control the occlusion ratios in two sensors' views, we generated our own datasets to evaluate the performance of our proposed algorithm for heavily occluded situations. A turntable was used to obtain ground truth. Several objects were placed on the center of the turntable, and the images were captured by a Kinect mounted on a tripod. We generated our dataset from the two scenes illustrated in Fig. 3.5 and in each scene the turntable was rotated clockwise incrementally at intervals of 5° up to 90°.

The main difference with this simulation in comparison to the previous set is that the ground truth was known exactly at every 5° interval, which was precisely

(a) Experimental setup (b) Scene 1 (c) Scene 2

Figure 3.5: Experimental setup and two scenes with different occlusions.

controlled, whereas in the previous simulation, the motion between two depth images was quite random. In this simulation, we could determine when the algorithms fail to provide the accurate estimation. The performance of the different algorithms was evaluated based on the rotational and translational residual mean square error (RMSE), as illustrated in Fig. 3.6.

The graphs in Fig. 3.6 clearly indicate that

- When the angular interval is greater than 15 degrees, more occlusion exist between two sensors' views. The proposed algorithm outperforms other variants, as it reports much lower translational and rotational RMSE.
- Standard ICP has the poorest performance across the experiments. ICP-IVD can provide similar accuracy in pose estimation before it diverges. However, as the scene becomes more occluded as the turntable is rotated, ICP-IVD fails to converge sooner than our proposed method.
- ICP-BD diverges at larger turntable rotation degrees than ICP and ICP-IVD. This indicates ICP-BD can deal with heavier occlusions between two views.
- For small angular intervals, the relative accuracy between the three algorithms are small. Therefore, all can be used to handle small motion in the presence of minimal occlusion.

A successful point cloud alignment after depth image registration using ICP with the bidirectional beam model is illustrated in Fig. 3.7.

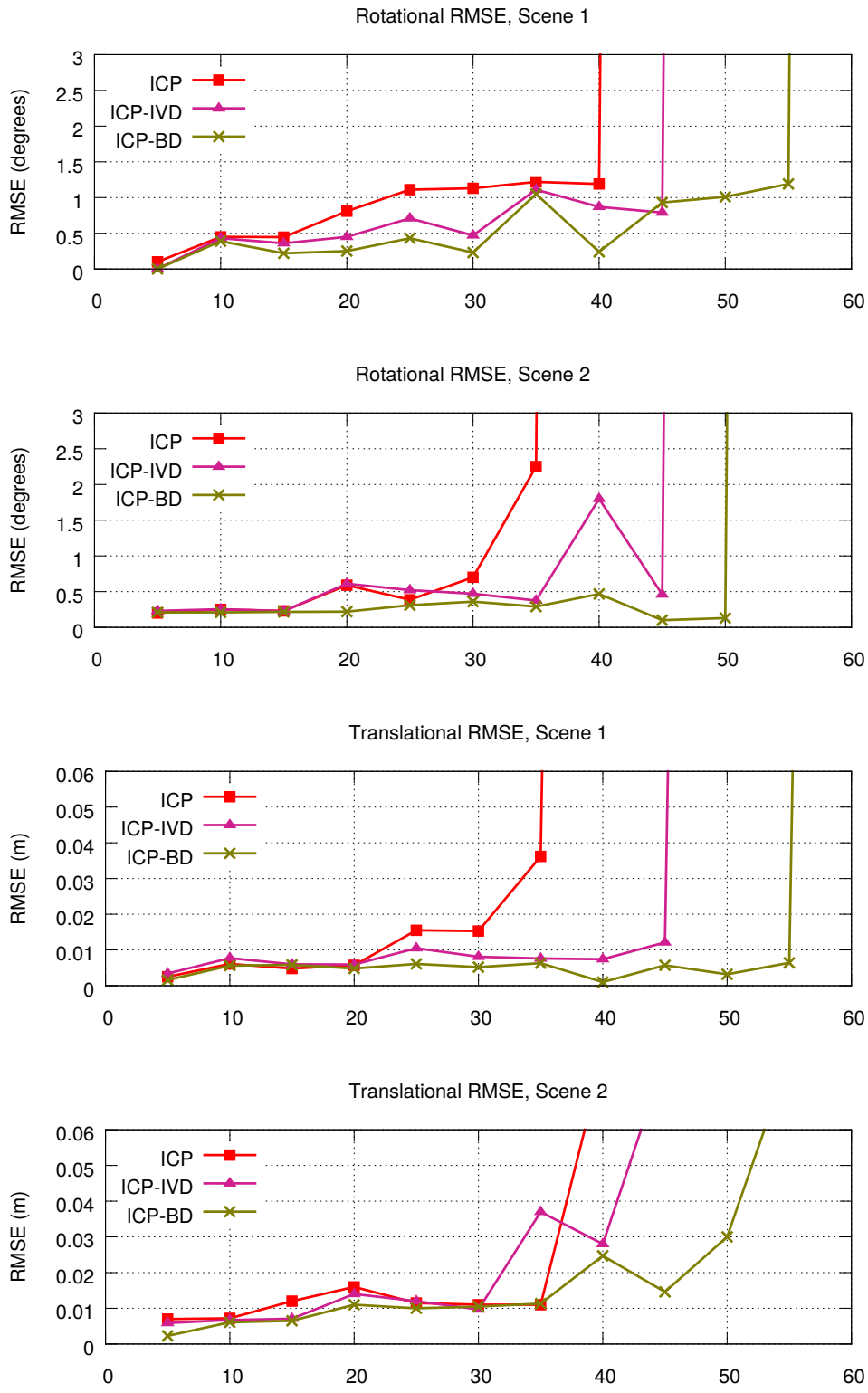


Figure 3.6: Comparison of the rotational and translational RMSE values for the ICP, ICP-VD and ICP-BD algorithms over two scenes shown in Fig. 3.5. X-axis values of the graphs show the amount of turntable angular rotation between two consecutive frames in degrees.

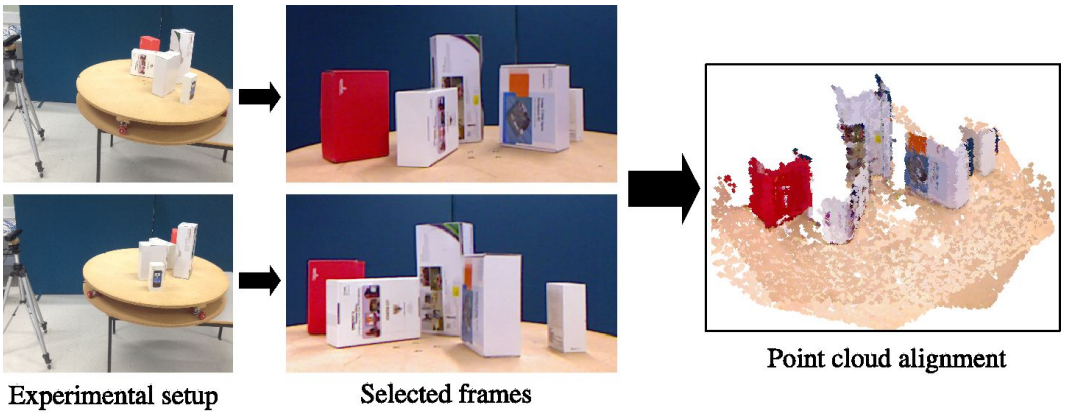


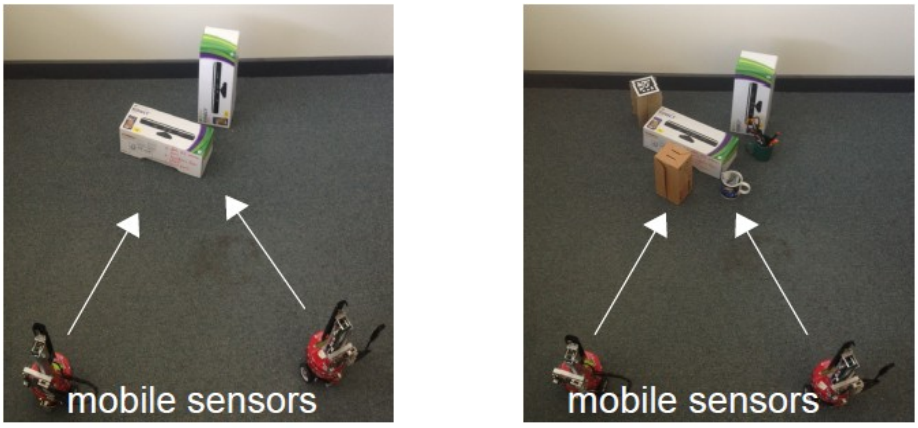
Figure 3.7: Point cloud alignment after depth image registration using ICP with bidirectional beam model. Two frames are from "Scene 1" with 50^o in turntable rotation interval. In this case, the other algorithms fail to provide correct relative pose estimation.

3.5.3 Mobile Visual Sensor Network Testbed Experiments

In this set of experiments, we implemented the proposed algorithm on our mobile visual sensor network testbed. The algorithm ran on the BeagleBoard-xM single-board computer installed on each mobile sensor in the network distributively.

We generated two different scenes illustrated in Fig. 3.8. In the following experiments, we performed 50 trials per scene. In each trial, we placed two sensors at different locations while maintaining their views of the scene from different angles. Two eyeBugs are able to communicate with each other directly through the wireless channel. The proposed algorithm was implemented on each eyeBug and worked in a distributed manner. As we did not have the precise ground truth information of each eyeBug's location and orientation in this set of experiments, we programmed the first eyeBug to keep stable and programmed the second eyeBug to move to the first eyeBug's position after it obtained the relative pose information. We deemed a trial to be successful if the second eyeBug moved to within 10 centimeters of the first eyeBug's position.

In Fig. 3.9 we present the frequency of successful trials where one eyeBug moved to the other eyeBug's position. When the amount of occlusion and clutter increases, our algorithm performs 10% to 16% better than ICP-IVD and much better



(a) Scene 1 (b) Scene 2

Figure 3.8: Two scenes with varying amounts of occlusion and clutter.

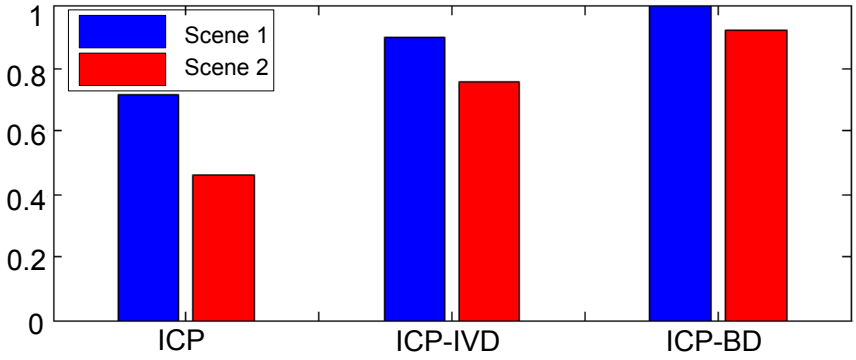


Figure 3.9: Frequency of successful relative pose estimation.

than the benchmark ICP. Due to the computational constraint of our mobile RGB-D sensors, the algorithm requires an average of 1.21 seconds to provide the relative pose estimation.

3.6 Summary

In this chapter, we propose the first relative pose estimation algorithm for multiple RGB-D sensors. The proposed approach can operate in real time on depth images captured by Microsoft Kinect. It has been implemented on a mobile visual sensor network to enable a sensor to obtain the location and orientation information of

the other sensor in the network.

The main contribution of this chapter is the development of a novel maximum likelihood model, named the bidirectional beam model, which can deal with the effect of occlusion in the views of different sensors. We incorporated this model into the ICP framework in order to determine the motion parameters. Different from the existing centralized algorithms, the proposed algorithm operates distributively on two sensors. This characteristic makes the proposed algorithm suitable for relative pose estimation. We conducted three sets of experiments to evaluate the accuracy and robustness of our proposed algorithm in environments with various amounts of occlusion. The results of the experiments indicate that the proposed ICP with bidirectional beam model is robust and accurate in different environments for relative pose estimation.

CHAPTER FOUR

SELF-CALIBRATION FOR RGB-D CAMERA-EQUIPPED VSNS

While the previous chapter presented a method to estimate the relative pose between two RGB-D sensors, this chapter focuses on using this relative pose estimation algorithm to calibrate VSNs equipped with $N \geqslant 3$ RGB-D cameras. We have developed a self-calibration algorithm to achieve this goal. We first model a VSN as an edge-weighted graph. Then, based on this model, and using real-time color and depth data, the sensors with shared FoVs estimate their relative poses in pairwise. The system does not need the existence of a single common view shared by all sensors, and it works in 3D scenes without any specific calibration pattern or landmark. Since proposed scheme distributes working loads evenly in the system, it is scalable and the computing power of the participating sensors is efficiently used. Sections of this chapter have been published in a conference paper [WŞD15a] and a journal paper [WŞD14].

4.1 Introduction

Estimating the geometry of a VSN from captured image information only, i.e. self-calibration [SHJH08, KS11], is a prerequisite for collaborative tasks. Self-calibration algorithms simultaneously process several images captured by different cameras and find the correspondences across images. Correspondences are established by

extracting 2D features from images automatically and matching them between different images. Then, based on the established correspondences, cameras' relative poses can be estimated from the essential matrix [HL06, LF06, RHH08, JSF12, ABS08]. The accuracy of the self-calibration greatly depends on the reliability of the relative pose estimates. This problem was first discussed in [DR04] with the concept of the vision graph. Kurillo *et al.* [KLB08], Cheng *et al.* [CDR07], and Vergés-Llahí *et al.* [VLMW08] later used and refined it for this purpose. It is becoming a useful general tool for describing the directionality of networked visual sensors. This approach has been more recently addressed by Bajramovic *et al.* [BD08, BBD12, BBD14]. They proposed a graph-based calibration method which measures the uncertainty of the relative pose estimation between each camera pair. *All of the self-calibration algorithms measure the epipolar structure of the system and suffer from scale ambiguity. If there is not any object or pattern with known geometry in the scene, the orientations and locations between cameras are determined up to a scale.*

Self-calibration for VSNs with conventional cameras is a well-developed area. However, to the best of our knowledge, no research work on self-calibration for multiple RGB-D sensors has been reported. In this chapter, we consider the self-calibration problem in a VSN with $N \geqslant 3$ RGB-D sensors operating in GPS-denied indoor environments (see Figure 1.3). Each sensor, named "eyeBug", has local processing ability with a RGB-D camera mounted on the top. A central node with a high performance processor is also implemented in the system, which can operate computationally expensive computer vision algorithms. We present a novel self-calibration algorithm to determine the locations and orientations of the sensors in this RGB-D cameras-equipped VSN. The proposed scheme can be arranged for indoor scenarios without the constraint for all sensors to share a common FoV. Our approaches assume that at least any two given sensors have overlapping FoVs and that the RGB-D cameras on sensors have been internally calibrated prior to deployment. Our proposed algorithms consist of the following steps:

1. each sensor extracts color features locally and sends the descriptors of these

features to the central node,

2. the central node performs feature matching to determine neighboring sensors and generates an Initial Pose Matrix (IPM),

3. the central node constructs a sensor dependency graph and selects a number of relative poses to connect sensors as a calibration tree,

4. after the central node broadcasts the information of the calibration tree, sensors work collaboratively to determine the relative poses according to the calibration tree,

5. the determined relative poses are then transmitted to the central node to compute the poses of all the sensors in the system.

We formulated the selection of relative poses as a shortest path problem, which consists of finding the shortest path from a vertex to the other vertices in an edge-weighted graph. The graph represents the FoVs of sensors as vertices and overlapping FoVs as edges, respectively.

The main contributions of this chapter are:

- Construction of the sensor-dependency graph based on the overlapping ratio between neighboring sensors.

- In contrast to the conventional approaches that utilize only color information, our approach takes advantage of the combination of RGB and depth information.

- The locations and orientations of sensors are determined up to the real world scale directly without involving scale ambiguity.

4.2 Self-Calibration Algorithm

4.2.1 Overview

Given $N(N \geqslant 3)$ sensors equipped with intrinsically calibrated RGB-D cameras, the goal is to automatically determine the initial pose of each sensor in a common coordinate system using only the color and depth data. A central node with a high performance processor is also included in the system to runs the computationally expensive algorithms.

When two sensors a and b have sufficiently overlapping FoVs, the relative pose between two sensors can be represented by a transformation matrix, $\mathbf{M}_{ab}$, in SE(3) as follows

$$\mathbf{M}_{ab} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. $\mathbf{M}_{ab}$ denotes the relative pose of Sensor b with respect to Sensor a and is the rigid transformation from the coordinate system of Sensor b to that of Sensor a. If there is a Sensor c and the relative pose between sensors c and b is $\mathbf{M}_{bc}$, then the relative pose between sensors a and c can be derived via composition as,

$$\mathbf{M}_{ac} = \mathbf{M}_{bc}\mathbf{M}_{ab} \tag{4.2}$$

This transformation provides a mapping from the coordinate system of c to that of b, then from that of b to that of a. Sensor b is the *intermediate node* in this process. This operation is transitive, therefore one sensor's pose relative to another can be determined indirectly over an arbitrary number of intermediate poses if they exist.

Therefore, the system's topology can be built up from the pairwise relative poses between sensors that have common FoVs. In order to achieve this, we first need to determine the sensors with sufficiently overlapping FoVs. Secondly, sensors are

grouped in pairs to determine rough estimations of the relative poses, and a number of relative poses are selected based on the reliability of the pose information. In the final step, we calibrate the overall system based on the selected pairwise relative poses. A general description of the scheme we propose is shown in Figure 4.1. Each step is described in detail in the following sections.

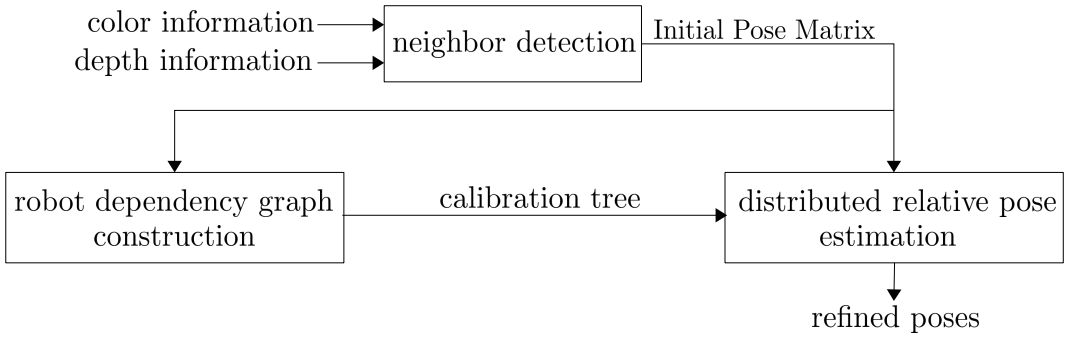


Figure 4.1: Operational overview of the proposed self-calibration scheme for cooperative pose estimation.

4.2.2 Assumptions

We make the following assumptions about the visual sensor network:

- Intrinsic parameters of the RGB-D camera on each sensor are calibrated prior to deployment,
- Each sensor in the system has overlapping FoV with at least one other sensor;
- The scene is static and the sensors do not move during the localization process, and
- The sensors can form an ad-hoc network and directly communicate with each other.

4.2.3 Neighbor Detection and Initial Relative Pose Estimation

We define sensors with overlapping FoVs as neighbors. One sensor's neighbors can be detected by searching for image pairs sharing common FoVs. This search can be viewed as a matching of point correspondences that considers the local environment of each feature point set. There are three steps in neighbor detection:

feature detection, feature description, and feature matching. The first two steps are performed on each sensor locally. Taking processing speed and accuracy into consideration, we implement FAST [RD06b, RD06a] for feature detection and ORB [RRKB11] for feature description on each sensor. Instead of transmitting the complete images, each sensor sends the feature descriptors to the central node to minimize the transmission load. The corresponding depth information of each feature is also transmitted in conjunction with the feature descriptors.

Associating the feature descriptors with their corresponding depth values, the central node can generate feature points in 3D. The central node performs feature matching between every two sets of the feature descriptors. In order to increase the matching reliability and robustness against outliers, we adopt both the symmetrical matching scheme and the geometric constraint to refine the matched points. In the symmetrical matching scheme, the correspondences between two sets of feature descriptors are established bidirectionally. One group of correspondences is generated from matching the first feature set to the second feature set. The other group is produced from matching the second feature set to the first feature set. For a pair of matched features to be accepted, two features must be the best matching candidates of each other in both directions.

Then, we use RANSAC to find a coarse registration, $\mathbf{M}_{ij}^{*}$, between every two matched feature sets. The error metric used to find the best alignment is

$$\mathbf{M}_{ij}^{*} = \underset{\widetilde{\mathbf{M}}_{ij}^{*}}{\arg\max} \left(\sum_{l=1}^{n} |\widetilde{\mathbf{M}}_{ij}^{*} \mathbf{p}_{i}^{l} - \mathbf{p}_{j}^{l}|^{2} \right) \tag{4.3}$$

where, $\mathbf{p}_{i}^{l}$ and $\mathbf{p}_{j}^{l}$ contain the depth information of two matched feature points as described in Equation 3.4. Each term in the summation indicates the squared distance between the transformed pose of a feature point $\mathbf{p}_{i}^{l}$ in Sensor i's feature set and the matched feature point $\mathbf{p}_{j}^{l}$ in Sensor j's feature set. Between every two matched feature sets, the central node samples a number of matched feature point pairs and determines the transformation matrix repeated. The determined

transformation in each iteration is evaluated based on the number of inliers in the remaining 3D feature points. Ultimately, only the matched feature points which agree with the optimal transformation matrix are kept as good matches. The determined coarse registrations between every two matched feature sets are stored as the *initial relative poses*. Initial relative poses are not accurate and require further refinements.

After operating the above process on every two feature sets, an Initial Pose Matrix (IPM) can be constructed. As shown in Table 4.1, each element, $\mathbf{M}^*_{ij}$, represents the initial relative pose between Sensor i and Sensor j. Since the diagonal elements represent the relative pose with itself, they are negligible.

Table 4.1: Initial Pose Matrix (IPM) and Uncertainty Matrix (UM) of a VSN with four sensors.

FMM					**UM**				
No.	**1**	**2**	**3**	**4**	**No.**	**1**	**2**	**3**	**4**
1	$\times$	$\mathbf{M}^*_{12}$	$\mathbf{M}^*_{13}$	$\mathbf{M}^*_{14}$	1	$\times$	w_{12}	w_{13}	w_{14}
2	$\mathbf{M}^*_{21}$	$\times$	$\mathbf{M}^*_{23}$	$\mathbf{M}^*_{24}$	2	w_{21}	$\times$	w_{23}	w_{24}
3	$\mathbf{M}^*_{31}$	$\mathbf{M}^*_{32}$	$\times$	$\mathbf{M}^*_{34}$	3	w_{31}	w_{32}	$\times$	w_{34}
4	$\mathbf{M}^*_{41}$	$\mathbf{M}^*_{42}$	$\mathbf{M}^*_{43}$	$\times$	4	w_{41}	w_{42}	w_{43}	$\times$

4.2.4 Selection of Relative Poses

After determining the neighboring sensors and initial relative poses, we show the problem of estimating all sensors' poses can be transformed to the all-pair shortest path problem.

The relative pose between two neighboring sensors can be estimated using the relative pose estimation algorithm. In order to calibrate the whole system, we need to select a number of relative poses to link all sensors together, since different overlapping areas in FoVs lead to various uncertainty values in relative pose estimation.

This process should choose the relative poses with the minimum overall uncertainty between two sensors. Furthermore, it is known that the accuracy of the estimation of the relative pose between two sensors may significantly degrade when increasing numbers of intermediate nodes are added to the computations. This is mainly due to the uncertainty accumulated each time the relative pose estimation algorithm operates between two sensors. In order to ensure each sensor has reliable knowledge of the other sensors' locations and orientations, we need to select the relative poses which introduce the smallest overall amount of uncertainty value to calibrate the system.

A. Sensor Dependency Graph Construction

To efficiently consider all possible combinations of sensor poses, we suggest the usage of the graph structure *sensor dependency graph*. A sensor dependency graph consists of a set of vertices representing each view of the scene observed by a sensor. The weight on each edge indicates the degree of uncertainty of the pair of views being connected. Thus, estimating all sensors' poses can be transformed to finding the shortest path between every two vertices in the sensor dependency graph.

In order to determine the weight on each edge, we need to first derive the uncertainty degree of relative pose estimation between every two neighbors. The relative pose between two neighboring sensors can be estimated by aligning the 3D point clouds extracted from the depth images captured by different sensors. We used our algorithm [WŞD13a] proposed in Chapter 3 to estimate the relative pose, since it reports more accurate and robust results in environments with various amount of occlusions than the current methods. The performance of the relative pose estimation algorithms depends on the overlapping area between two FoVs. In the same circumstance, a larger overlapping area leads to a better alignment and a more accurate estimate. The overlapping area between two neighbors can be estimated by the initial relative pose determined in Section 4.2.3.

Let $\mathbf{M}_{ab}$ denote the relative pose between sensors a and b. The pixels of the

depth image captured by Sensor a can establish a relation with their projections on the depth image captured by Sensor b as

$$\begin{bmatrix} u_b & v_b & 1 & q_b \end{bmatrix}^T = \mathbf{M}_{ab} \begin{bmatrix} u_a & v_a & 1 & q_a \end{bmatrix}^T \tag{4.4}$$

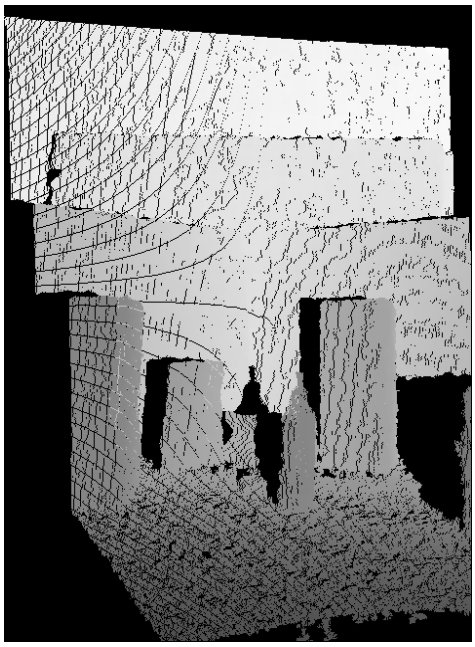
$$\begin{bmatrix} \frac{i_b - i_{c,b}}{f_{x,b}} & \frac{j_b - j_{c,b}}{f_{y,b}} & 1 & \frac{1}{z_b} \end{bmatrix}^T = \mathbf{M}_{ab} \begin{bmatrix} \frac{i_a - i_{c,a}}{f_{x,a}} & \frac{j_a - j_{c,a}}{f_{y,a}} & 1 & \frac{1}{z_a} \end{bmatrix}^T \tag{4.5}$$

Here, $[u_a, v_a, 1, q_a]^T$ indicates a 3D point in the inverse depth coordinate system, (i_a, j_a) and (i_b, j_b) are the pixel coordinates on different depth images, $(i_{c,a}, j_{c,a})$ and $(i_{c,b}, j_{c,b})$ are the principal points of cameras on two sensors, $(f_{x,a}, f_{y,a})$ and $(f_{x,b}, f_{y,b})$ are the focal lengths, and z_a and z_b are the depth values of the same real world point projections in different depth images. By applying Equation (4.5) on the depth image observed by Sensor a, we can generate a synthetic view which is virtually taken at sensor b's viewpoint. The overlapping area between two sensors' FoVs can be determined through comparing the real and synthetic depth images. We define the *overlapping ratio* between two neighbors as the proportion of overlapping area in the observed image. However, in this approach the central node requires the knowledge of the complete depth image of Sensor a. If all the sensors have to transmit their observed depth images to the central node, a considerable transmission load will be generated. In order to efficiently estimate the overlapping ratio, Sensor a can only send the values and coordinates of four pixels in its observed depth image. These four pixels are the nearest pixels with valid depth values to the four corners (top left, top right, bottom left, and bottom right) of a image. After applying Equation (4.5) on these four pixels, the quadrangle constructed by the reprojections of these four pixels indicates the region observed by Sensor a in Sensor b's view. Although the points in the scene lie on different planes and have various range values, this approach can still provide a rough estimate of the overlapping ratio. An example is shown in Figure 4.2.

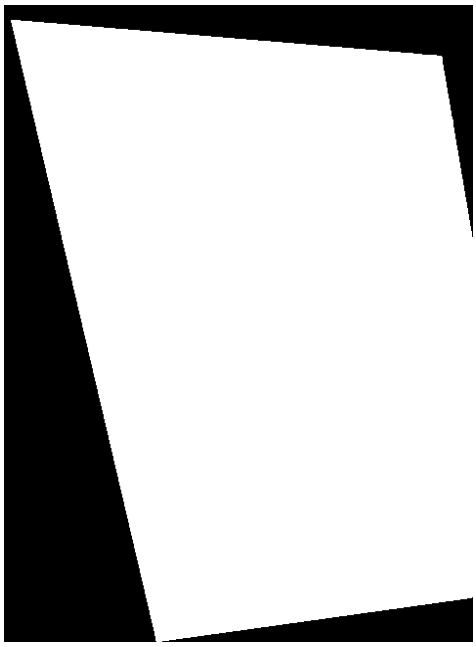
The relation between the overlapping ratio and the uncertainty in the results

(a) Robot a's view

(b) Robot b's view



(c) Synthetic view of robot b



(d) Estimated overlapping area

Figure 4.2: Overlapping area estimation. (**a**) Depth image captured by Sensor a, 4 corner pixels are highlighted in red; (**b**) Depth image captured by Sensor b; (**c**) Depth image virtually captured at sensor b's position generated from (**a**); (**d**) The rough estimate of overlapping area in Sensor b's view. The white region indicates the overlapping region.

provided by the relative poses estimation algorithm [WŞD13a] were explored by analyzing the results of the experiments reported in Chapter 3.5.1. We grouped image pairs with various overlapping ratios and determined the average error in relative location estimation (since error in orientation estimation is hard to evaluate quantitatively, we considered the error in translation only). The results are presented in Table 4.2.

Table 4.2: Average relative error in the estimated relative location between two sensors with different overlapping ratios.

Overlapping ratio (ϕ_{ij})	Relative error in location
$\phi_{ij} \geqslant 0.7$	0.8%
$0.7 > \phi_{ij} \geqslant 0.6$	1.2%
$0.6 > \phi_{ij} \geqslant 0.5$	1.9%

By assuming the error in relative pose estimation accumulates linearly, the equation

$$w_{ij} = \begin{cases} 1 & \text{if } \phi_{ij} \geqslant 0.7 \\ 1.5 & \text{if } 0.7 > \phi_{ij} \geqslant 0.6 \\ 2.4 & \text{if } 0.6 > \phi_{ij} \geqslant 0.5 \\ \infty & \text{if } 0.5 > \phi_{ij} \end{cases} \tag{4.6}$$

is adopted to quantize the overlapping ratio and uncertainty degree. Here, ϕ_{ij} represents the overlap ratio between two sensors i and j, and w_{ij} indicates the uncertainty degree in relative pose estimation between two neighboring sensors. A larger w_{ij} indicates a larger uncertainty value in the relative pose estimation. Based on the IPM and the criteria in Equation (4.6), the Uncertainty Matrix (UM) can be generated.

According to the UM, we can generate the sensor dependency graph, $G = (V, A)$. There is an edge between any two neighboring sensors iff the overlapping ratio is within the range in Equation (4.6). The weight of the edge linking sensors i and j is w_{ij}, which indicates the uncertainty degree. A lower w_{ij} indicates a smaller uncertainty value in relative pose estimation. Then, the problem of relative

pose selection is transformed to the all-pairs shortest path problem in the sensor dependency graph, which minimizes the uncertainty in the relative pose estimation between every two sensors. To the best of our knowledge, we propose the first sensor dependency graph which uses the overlapping ratios between FoVs as the weights of the edges.

B. Calibration Tree Construction

The shortest path between every two vertices can be determined using the Floyd–Warshall algorithm [CSRL01]. The central node first generates *Dist* as a $|V| \times |V|$ array of minimum distance and initializes *Dist* according to UM. Next, the Floyd–Warshall algorithm is used to determine the shortest paths between every pair of sensors and update *Dist*. Then, the central node needs to select one sensor as the *primary sensor* and make all the other sensors calibrate their poses according to the primary sensor's coordinate system. In order to minimize the uncertainty, the central node selects the sensor which has the smallest overall weight on the shortest paths to all the other sensors as the primary sensor. Finally, the sensors can be connected as a *calibration tree* with the primary sensor as the root. In this method, the relative pose estimation algorithm only requires to operate $|V| - 1$ times to connect all the sensors. The time complexity of the overall scheme is $O(V)$. Therefore, this scheme is scalable to initialize VSNs with a large number of RGB-D sensors.

4.2.5 Distributed Relative Pose Estimation Algorithm

Although the initial relative poses on the edges of the calibration tree have already been obtained in the neighbor detection process, these estimations are not accurate enough to calibrate the overall system. Therefore, after the calibration tree is built, the central node will broadcast the connection information of the calibration tree and the related initial relative poses to all the sensors. Then, the relative pose estimation algorithm described in Chapter 3 will operate on each sensor locally

to compute the relative poses in the calibration tree. Different from the methods for conventional RGB cameras which use feature correspondences to determine the rotation and translation up to a scale, this distributed, peer-to-peer algorithm determines the relative pose in consistent real world scale through the explicit registration of surface geometries extracted from two depth images. The registration problem is approached by iteratively minimizing a cost function in which error metrics are defined based on the bidirectional point-to-plane geometrical relationship [WŞD13a]. However, instead of using identity transformation as the initial guess, we use the initial relative pose determined in the neighbor detection process as the initial guess for relative pose estimation.

The relative poses that construct the calibration tree are transmitted to the central node after being determined by sensors. Then all sensor locations and orientations can be calibrated according to the primary sensor's coordinate system. A simple example of the working process is shown below. Figure 4.3 depicts a calibration tree for initializing a VSN with 4 sensors. Sensors operate the relative pose estimation algorithm to derive the relative poses $\mathbf{M}_{ab}$, $\mathbf{M}_{ac}$, and $\mathbf{M}_{cd}$ according to the tree. By using these three pose matrices, the relative pose between every two sensors in the network can be derived. For instance, Sensor d's location and orientation in Sensor b's coordinate system can be derived as

$$\mathbf{M}_{bd} = \mathbf{M}_{ad}\mathbf{M}_{ba} = \mathbf{M}_{cd}\mathbf{M}_{ac}\mathbf{M}_{ab}^{-1} \tag{4.7}$$

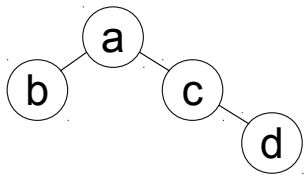


Figure 4.3: Example of a calibration tree.

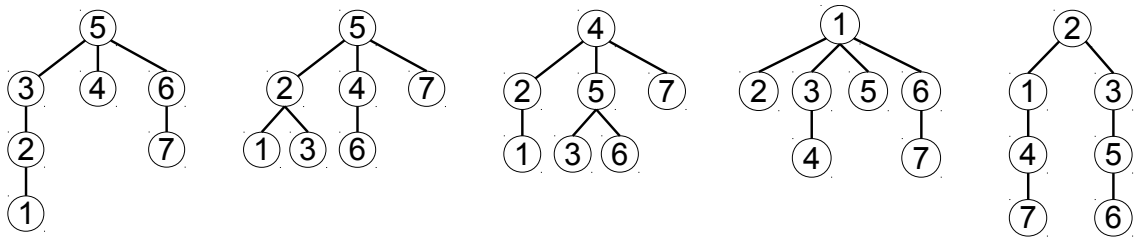
4.3 Experimental Results and Discussion

In order to quantitatively evaluate the performance of the proposed method, we conducted a series of experiments, both in simulation and with our custom-built VSN [1]. Section 4.3.1 describes the localization experiments that were carried out with our experimental VSN in indoor environments. Section 4.3.2 presents the results of a set of simulations designed to further evaluate the behavior of the method.

4.3.1 Indoor Experiments

We used the color and depth images captured by our experimental VSN consisting of seven RGB-D sensors. The images were taken from various locations in and around the WSRNLab facility [wsr]. Color images of collected five scenes are shown in Figure 4.5. As the sensors were deployed on the same plane in this set of experiments, the ground truth locations and orientations could be easily measured manually. The estimated sensors' poses are shown in Figure 4.6, in which the estimated poses are depicted in red circles and the ground truths are represented in blue stars. We derived the average absolute errors accordingly and the results are presented in Table 4.3. The calibration trees of 5 scenes are illustrated in Figure 4.4. The pose estimations of Scene 1 have the smallest absolute error, while the estimates in Scene 5 have the largest absolute error. We also measured the average relative error for localization. The relative errors were computed based on the absolute errors and the system dimensions. It is clear that the pose estimation results in Scene 4 and Scene 5 are the most and least accurate of the five scenes respectively. By analyzing the sensors' sensing ranges in different scenes, we found that sensing range is the main factor that affects the performance of our proposed scheme. As reported in [Kho11], the errors in the depth measurements of Kinect

[1] As there is not any self-calibration algorithm for RGB-D sensors and self-calibration algorithms for conventional cameras suffer from the scale ambiguity problem, we cannot compare the performance of our proposed method with the performance of other approaches.



(a) Scene 1 (b) Scene 2 (c) Scene 3 (d) Scene 4 (e) Scene 5

Figure 4.4: Calibration trees of indoor experiments.

Table 4.3: Average error between the estimated poses and the ground truth.

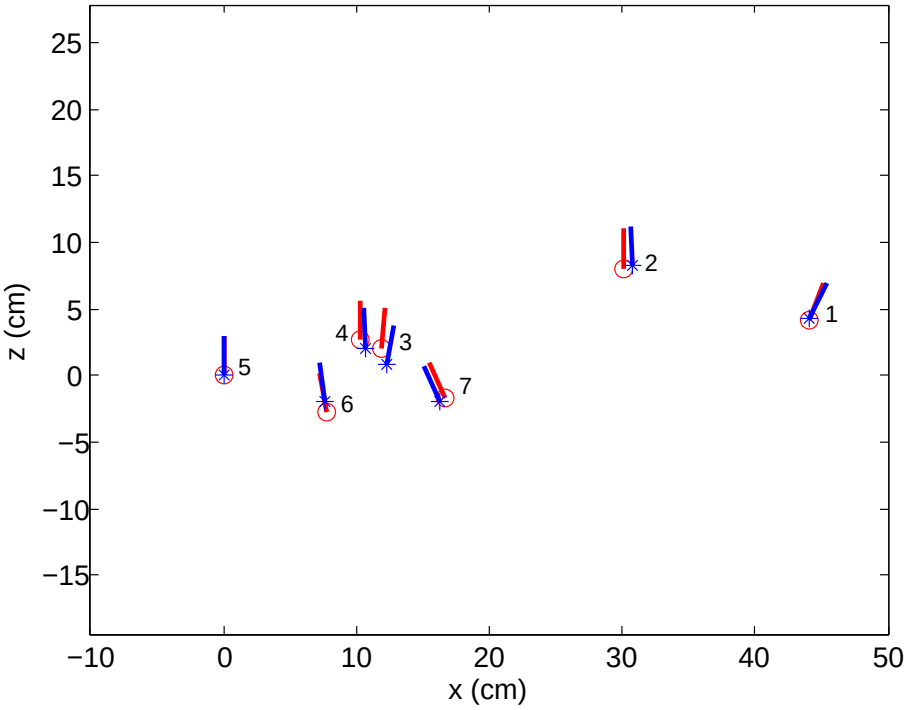
Data Set	Sensing Range		Average Absolute Error		Localization Average Relative Error
	Max (m)	Average (m)	Location (mm)	Orientation (°)	
1	1.92	1.47	10.0	1.6	2.26%
2	6.23	1.72	14.8	2.3	1.36%
3	3.95	1.86	25.1	2.7	1.39%
4	1.79	1.41	12.6	2.1	1.12%
5	6.02	4.14	64.7	6.2	3.81%

increase quadratically from a few millimeters at 0.5 m distance to about 4 cm at the maximum range of the sensor. The inaccurate depth information obtained by the Kinect on each sensor influences the performance of the distributed relative pose estimation algorithm, thereby decreasing the accuracy of the overall scheme. Due to this limitation of the RGB-D camera, the sensors' sensing ranges should be controlled to between around 0.5 m to 3.5 m.

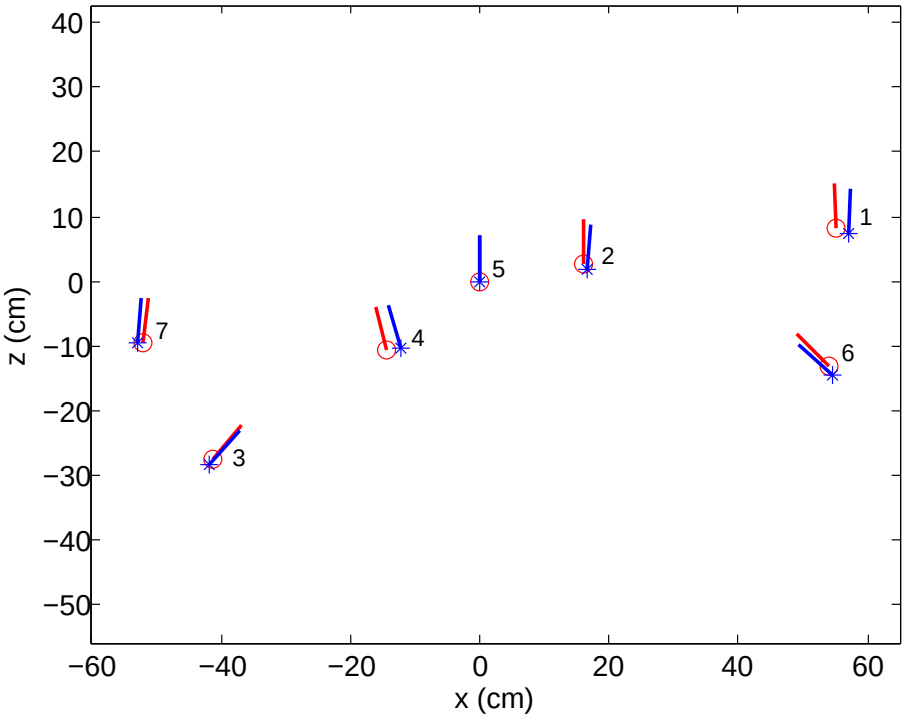
4.3.2 Simulation Experiments

A series of simulation experiments was conducted to investigate the accuracy and bandwidth consumption of the proposed scheme implemented on systems that were larger and have more complicated topologies than those we could construct with our available hardware. When sensors are deployed on different planes, the ground truth poses are difficult to measure precisely using manual methods. Therefore, in this set of simulations we applied 3D image warping technique [MFFT08] on the color and corresponding depth images captured in Scene 4 to generate synthetic image sets with known transformation matrices. Image sets were generated for systems consisting of 10, 15 , 20, 25, 30, 35, and 40 sensors. In this process, we ensured that each sensor had sufficiently overlapping FoV with at

Figure 4.5: Color images captured by the multi-sensor system in 5 representative scenes.

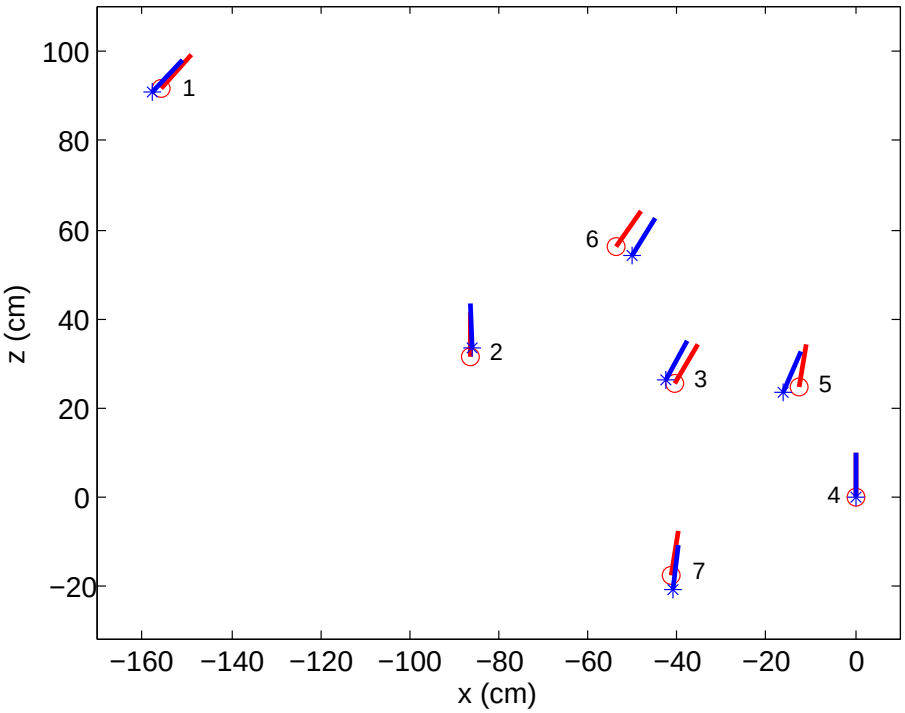


(a) Scene 1

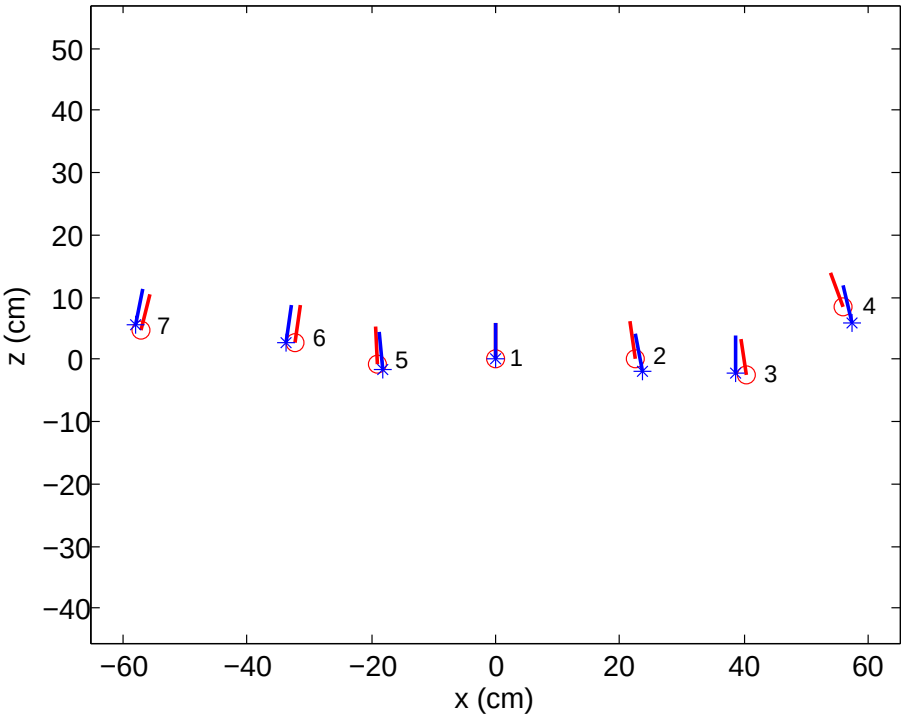


(b) Scene 2

Figure 4.6: Estimated and ground truth sensor poses in Scenes 1 and 2 (see Figure 4.5). Estimated locations are depicted by red circles and their ground truth positions are shown by blue stars. The line segments on different markers indicate orientations.



(a) Scene 3



(b) Scene 4

Figure 4.7: Estimated and ground truth sensor poses in Scenes 3 and 4 (see Figure 4.5). Estimated locations are depicted by red circles and their ground truth positions are shown by blue stars. The line segments on different markers indicate orientations.

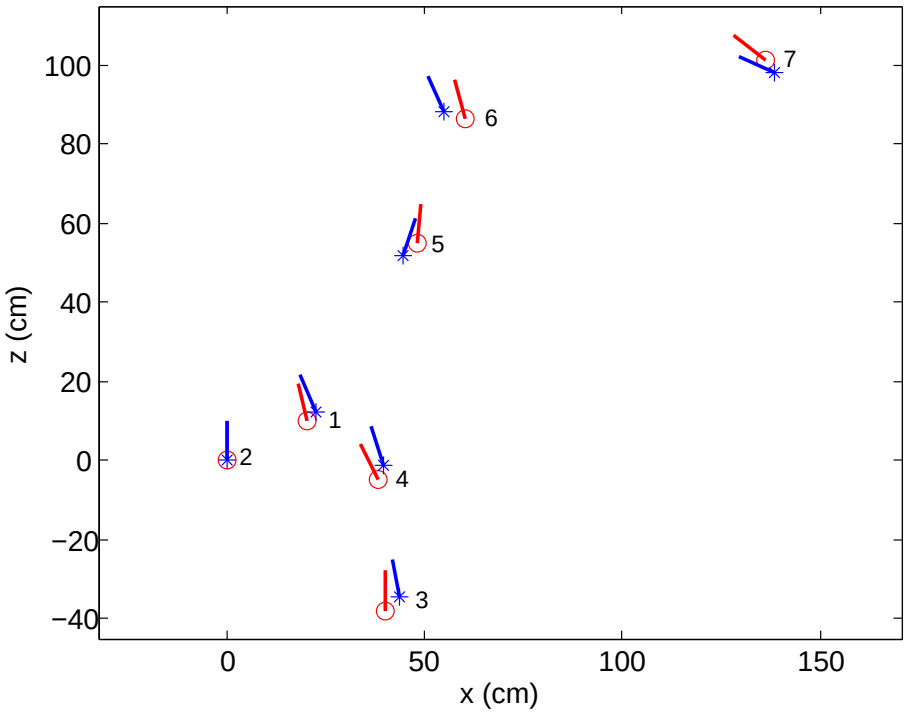
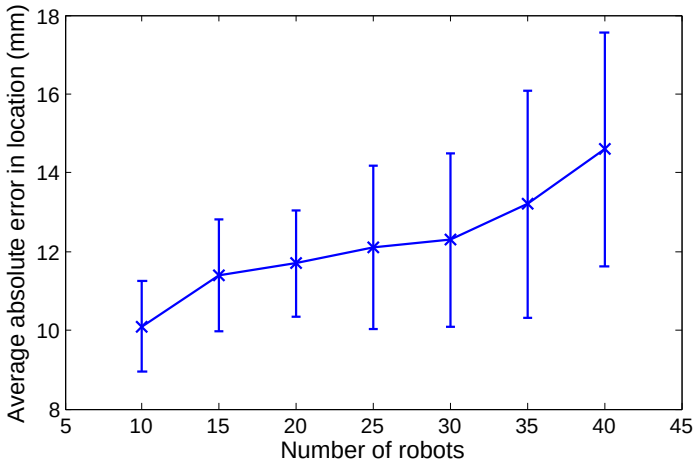


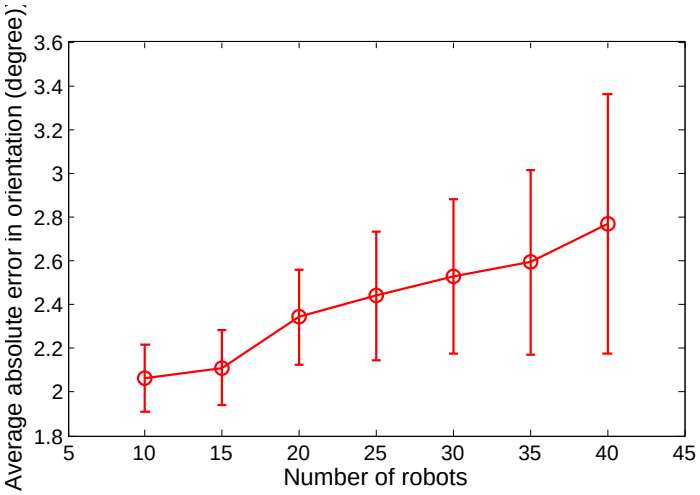
Figure 4.8: Estimated and ground truth sensor poses in Scene 5 (see Figure 4.5). Estimated locations are depicted by red circles and their ground truth positions are shown by blue stars. The line segments on different markers indicate orientations.

least one another sensor and could be connected in the calibration tree. The results presented in Figure 4.9 are averaged over 10 runs of the simulations with vertical bars indicating the variance.

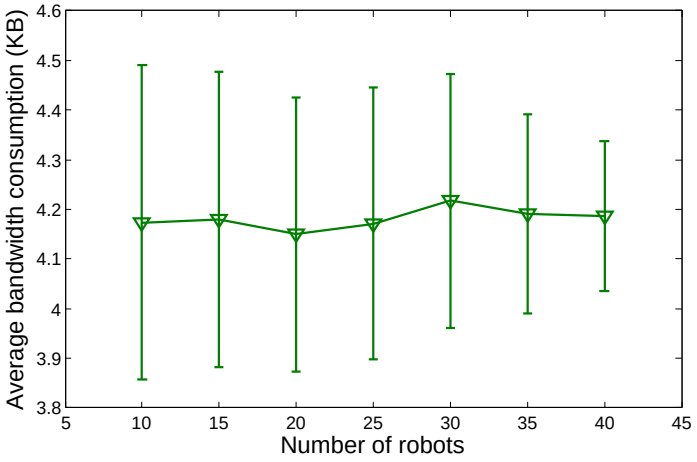
Figures 4.9a and b, indicate that the absolute errors in both location and orientation increase as the number of sensors in the system grows. This is because, when the number of sensors rises, the calibration tree becomes larger and the primary sensor requires more intermediate nodes to establish the connection with another sensor in the system. Although this effect accumulates small errors, the average absolute errors are still quite small and the average relative error is within 1.3%. The average bandwidth consumption of each sensor is presented in Figure 4.9c. As expected, the bandwidth usage per sensor remains consistent approximately in relation to the number of sensors in the system. Two processes in our proposed scheme require communication between sensors: (1) neighbor detection; and (2) distributed relative pose estimation. As the number of times that distributed relative pose estimation runs increases linearly with the number of sensors, each



(a) Average absolute error in location estimation with increasing number of sensors.



(b) Average absolute error in estimated orientation with increasing number of sensors.



(c) Bandwidth consumption per sensor with increasing number of sensors.

Figure 4.9: Simulation results showing the location estimation and average absolute errors, and bandwidth consumption as the number of sensors in the system increases.

sensor's transmission load in this process will not be influenced by the number of sensors in the system. The only variable that affects the transmission load is the number of feature points in the neighbor detection process. However, the number of feature points, which depends on the structure of the captured scene, is unrelated to the number of sensors. Therefore, the evenly distributed communication load throughout the system indicates the good scalability of our proposed scheme.

4.4 Summary

This chapter describes the first approach which uses color and depth information to initialize sensors' poses in a RGB-D camera-equipped VSN. Our scheme first detects each sensor's neighbors using robust feature matching. Then the overlapping areas between the FoVs of neighboring sensors are determined to establish the sensor dependency graph. A calibration tree is generated by finding the shortest path between the primary sensor to all the other sensors in the system. Finally, a distributed relative pose estimation algorithm is performed to precisely compute the relative pose between every two connected sensors in the calibration tree. Extensive real world experiments and synthetic dataset simulations have been conducted. The results show that our scheme is robust and accurate in different environments and with various densities of sensors. Importantly, the proposed scheme operates distributively and allows the sensors to use the limited wireless bandwidth more efficiently. This calibration algorithm, which provides initial location and orientation information, has great potential for use in a wide range of applications, such as visual SLAM and 3D reconstruction.

CHAPTER FIVE

EFFICIENT RGB-D DATA COMMUNICATION SCHEMES

This chapter focuses on using the sensors' pose information to achieve efficient RGB-D data communication in VSNs. In particular, we concentrate on the environments in which the communication bandwidth is severely limited and fluctuating. To this end, this chapter will first introduce a depth video compression scheme for a single mobile RGB-D sensor. Then, based on this algorithm, a collaborative color and depth data coding scheme will be presented for multiple sensors. Three publications [WŞD13c, WŞD$^+$15b, WŞD$^+$] have been generated from parts of this chapter.

5.1 Introduction

The invention of low-cost RGB-D cameras has made large-scale, high-resolution 3D sensing for mapping, immersive telepresence, surveillance, and environmental sensing tasks easily achievable. Furthermore, mobile robots with RGB-D sensors can form VSNs to work collaboratively and autonomously, reducing the time required to complete these tasks. However, the volume of visual and depth data generated by a VSN is large, which presents a challenge for efficient data transmission and storage, particularly over the shared wireless channels. The problem is further exacerbated by the operation of the robots in potentially hostile environments, such as mapping indoors after a disaster such as Fukushima nuclear

reactor accident and underground cave exploration, which leads to communication difficulties.

A literature review on methods to remove the redundant information in color and depth data is presented in Chapter 2. These schemes can be broadly classified into two categories: approaches developed to remove (i) inter-frame redundancies [GM04, HWDK08, DTPP09, SMAP14, KFMK09, FWL11, NMD13], or (ii) multi-view redundancies [ML05, CAS09, CCAS12, CCM12, WC07, ZHL10, SMW07, BAA06, EWK09, LWP11, Feh04, LCH11, Mar99, WHY11]. Inter-frame redundancies exist in consecutive frames captured by one sensor, while multi-view redundancies exist in images captured by neighboring sensors which have overlapping FoVs. Though many schemes have been proposed in recent years, those in the first category concern the exploration of the temporal correlation in the video captured by a fixed camera. The approaches in the latter category require the processor to have full knowledge of the images captured by all cameras or the pose differences between cameras are very small. Therefore, in this chapter, we address the redundancy removal problem in two circumstances. The first circumstance is typical in visual-SLAM (simultaneous localization and mapping) or surface reconstruction applications, in which a mobile RGB-D sensor collects depth and visual information of a static environment. In the second circumstance, multiple static RGB-D sensors have overlapping FoVs, while they have no knowledge about the images observed by the others. We propose a new method, called *3D Image Warping Based Depth Video Compression (IW-DVC)*, for fast and efficient compression of depth video captured in the first circumstance. Then, based on this method, we present the *Relative Pose based Redundancy Removal (RPRR)* scheme to efficiently remove the redundant information captured by each sensor before transmission.

This chapter is organized as follows. Section 5.2 presents the coding scheme for the depth video captured by a mobile RGB-D sensor. Section 5.3 presents the framework for removing the redundant color and depth information captured by multiple RGB-D sensors. The performances of two schemes are analyzed in Section

5.4. Finally, Section 5.5 summarizes the content of this chapter.

5.2 Depth Video Compression for a Mobile RGB-D Sensor

In the first circumstance, the depth video compression problem inevitably becomes much more complicated when moving cameras are involved. As the distance between a moving camera and the objects in a scene changes across time, depth values of the static objects change in successive depth frames. This characteristic is against the assumption of the most depth video coding schemes. These schemes assume the depth information of the static regions in the scene does not vary in successive depth frames. Therefore, the coding schemes with motion compensation methods for static cameras become very inaccurate or even useless with mobile cameras. As the RGB information can be compressed easily by conventional image/video coding schemes, we focus on the development of a fast and efficient coding method for the depth video.

5.2.1 System Overview

An overview of the new scheme we propose, called 3D *Image Warping Based Depth Video Compression* (IW-DVC), is shown in Fig. 5.1 and Fig. 5.2. We first eliminate the redundant depth data at the encoder side (Fig. 5.1). Let $\mathbf{Z}_I$ and $\mathbf{Z}_P$ denote an I-frame and P-frame in a group of pictures (GoP) in a captured depth video. The system consists of two main components: a motion compensation algorithm and a lossless coding scheme. Before encoding $\mathbf{Z}_P$ into a bitstream, the differences between $\mathbf{Z}_P$ and $\mathbf{Z}_I$ should be determined in motion compensation first. The first step of the encoding procedure (Fig. 5.1) is the inter-frame motion estimation. We estimate the motion of a moving sensor in the time interval of capturing depth frames $\mathbf{Z}_I$ and $\mathbf{Z}_P$. In the second step, $\hat{\mathbf{Z}}_P$, a prediction of $\mathbf{Z}_P$ is generated by using the interframe motion information, forward estimation/reverse check, and block-based update.

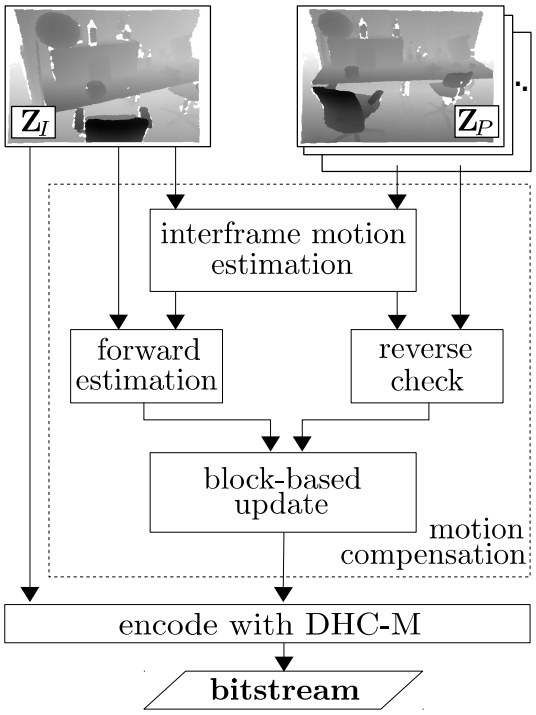


Figure 5.1: Encoding process of the *3D Image Warping Based Depth Video Compression* (IW-DVC) framework.

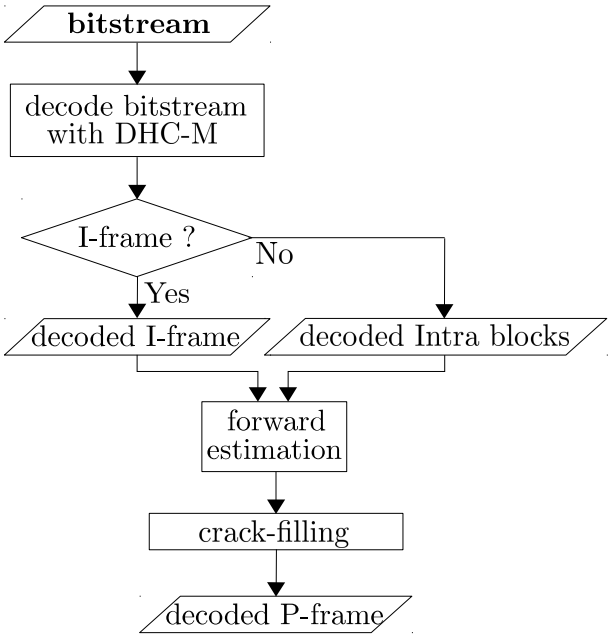


Figure 5.2: Decoding process of the *3D Image Warping Based Depth Video Compression* (IW-DVC) framework.

This procedure estimates the newly observed depth information in $\mathbf{Z}_P$. Then, only the newly observed information in $\mathbf{Z}_P$ is encoded using an entropy coding scheme. As a result, the redundancy in the depth video is removed during the encoding process.

In the decoding process, the received bitstream is decoded with the same codec used at the encoder side. I-frames can be directly decoded from the bitstream. Each P-frame has to be reconstructed using the decoded newly observed information (intra blocks) in each P-frame and its corresponding I-frame. Then, a crack-filling approach is used to deal with the under-sampling issue [MMB97] and enhance the quality of the reconstructed P-frames.

The proposed framework can be implemented directly on a mobile RGB-D sensor system to achieve efficient depth video storage and transmission. This approach enables better prediction of frames, leading to an enhanced compression performance by taking advantage of the characteristic properties of depth images. Furthermore, this motion compensation method is computationally efficient and can operate in real-time, since it does not need to derive the mean squared error (MSE) block-by-block, and it further can be used as a front-end for other depth image coding schemes to remove the redundancy between consecutive frames prior to further coding. The received depth information can be used for many applications, such as visual mapping and exploration, 3D scene reconstruction, or free viewpoint video rendering. The main contributions of this framework can be summarized as follows:

- Theoretical development and practical implementation of the first motion compensation algorithm for depth video captured by a mobile sensor/camera,
- Thorough performance evaluation of the proposed framework under the various parameter changes to the system, and
- Extensive experiments using multiple datasets to evaluate the coding performance under a variety of scenes.

5.2.2 Interframe Motion Estimation

Conventional 2D block-based motion estimation algorithms use 2D block matching approaches to estimate the MVs which can map the pixels from the reference frame to the current frame. However, this kind of approaches relies on the assumption that the pixels representing the surface information on the same object stay unchanged, even if the object or the camera is in motion. Therefore, it is clear that these 2D block-based motion estimation approaches are not optimal for depth videos, as the depth frames represent the distance of objects within a scene relating to the camera position and the depth values change when the camera or object is moving.

Taking advantage of the depth image characteristics, the depth pixels in the reference frame can be mapped to the current frame. We assume that a world point $\mathbf{p_e}$ can be observed in $\mathbf{Z}_I$ and $\mathbf{Z}_P$ captured by the mobile RGB-D sensor, and the projections of $\mathbf{p}_e$ are located at pixel coordinates (i_I, j_I) and (i_P, j_P) on the depth frames $\mathbf{Z}_I$ and $\mathbf{Z}_P$, respectively. Also, under the assumption that the world coordinates system is equivalent to the mobile sensor coordinate system, the depth pixel (projection) at (i_I, j_I) in $\mathbf{Z}_I$ can establish a relationship between the depth pixel at (i_P, j_P) in $\mathbf{Z}_P$ as follows,

$$\begin{bmatrix} \frac{i_P - i_c}{f_x} & \frac{j_P - j_c}{f_y} & 1 & \frac{1}{z_P} \end{bmatrix}^T = \mathbf{M} \begin{bmatrix} \frac{i_I - i_c}{f_x} & \frac{j_I - j_c}{f_y} & 1 & \frac{1}{z_I} \end{bmatrix}^T \tag{5.1}$$

To simplify the equation, by doing some rudimentary algebraic substitutions, we obtain the following equation in inverse depth coordinate,

$$\begin{bmatrix} u_P & v_P & 1 & q_P \end{bmatrix}^T = \mathbf{M} \begin{bmatrix} u_I & v_I & 1 & q_I \end{bmatrix}^T. \tag{5.2}$$

The transformation matrix $\mathbf{M}$ represents the 6 DoF motion model, which describes the motion of the sensor and the transformation of the structure between a pair of

depth images. It has the form

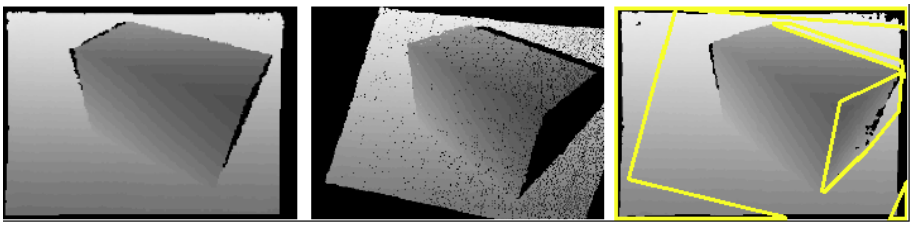
$$\mathbf{M} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. Therefore, if we have accurate information about the transformation matrix $\mathbf{M}$, each pixel in the reference depth frame can find its corresponding pixel in the current depth frame. We can treat depth frames $\mathbf{Z}_P$ and $\mathbf{Z}_I$ as the depth images captured by two RGB-D sensors, and the transformation matrix can then be derived using our proposed relative pose estimation algorithm, which is described in Chapter 3. It estimates $\mathbf{M}$ through the explicit registration of surface geometries extracted from two depth frames. The registration problem is approached by iteratively minimizing a cost function, the error metrics of which are defined based on the bidirectional point-to-plane geometrical relationship.

5.2.3 Forward Estimation/Reverse Check and Block-based Update

After determining $\mathbf{M}$, the correspondences between the depth pixels in $\mathbf{Z}_I$ and the depth pixels in $\mathbf{Z}_P$ are established by the *forward estimation/reverse check* and *block-based update* procedures. They are explained in this section.

A. Forward Estimation



$\mathbf{Z}_I$ $\hat{\mathbf{Z}}_P$ $\mathbf{Z}_P$

Figure 5.3: An intuitive example of forward estimation. The depth frame $\hat{\mathbf{Z}}_P$ is predicted from $\mathbf{Z}_I$ as the frame captured at P-frame's viewpoint virtually. After comparing $\hat{\mathbf{Z}}_P$ and $\mathbf{Z}_P$, the newly observed information in $\mathbf{Z}_P$ is outlined in yellow.

In the forward estimation, we try to generate a depth frame $\hat{\mathbf{Z}}_P$ which is the prediction of the depth frame $\mathbf{Z}_P$. We first initialize the pixels in $\hat{\mathbf{Z}}_P$ to the invalid depth value. Then, according to Eq. 5.1, each pixel in depth frame $\mathbf{Z}_I$ is warped to a coordinate in $\hat{\mathbf{Z}}_P$. In this process, it can happen that two or more different depth pixels are warped to the same pixel coordinate in $\hat{\mathbf{Z}}_P$. This over-sampling issue occurs because some 3D world points are occluded by others at the new viewpoint. We adopt Z-buffer [MFFT08] to solve this problem. By contrast, some regions in $\hat{\mathbf{Z}}_P$ may have no valid depth information, as none of the pixels in $\mathbf{Z}_I$ can be warped to these regions. We name these *hole regions* and pixels in them have invalid depth values. They indicate that some depth information in $\mathbf{Z}_P$ cannot be predicted from $\mathbf{Z}_I$. Therefore, the regions in $\mathbf{Z}_P$ with the same locations of hole regions in $\hat{\mathbf{Z}}_P$ are assumed to contain the newly observed depth information. An example of this process is shown in Fig. 5.3. In this example, the regions containing the depth information that can only be observed by $\mathbf{Z}_P$ are outlined in yellow.

B. Reverse Check

Although the forward estimation can detect the newly observed depth information in the current frame $\mathbf{Z}_P$ in most cases, it may fail to operate correctly in situations when some points are occluded by the objects that can only be seen in $\mathbf{Z}_P$. A typical scenario is shown in Fig. 5.4. In this example, as object A cannot be observed in I-frame, the forward estimation will falsely treat the background in red as the surface that can be observed in $\mathbf{Z}_P$. The surface of object A in green is observed in $\mathbf{Z}_P$ instead of the red background surface. Therefore, the forward estimation cannot accurately determine the newly observed depth information of $\mathbf{Z}_P$ in this case.

In order to solve this problem, we introduce a reverse check mechanism. Similar to the warping process in the forward estimation, in the reverse check process the pixels on depth frame $\mathbf{Z}_P$ can be warped to generate a depth frame, $\hat{\mathbf{Z}}_I$, which is captured at I-frame's viewpoint virtually. The warping process in the reverse check

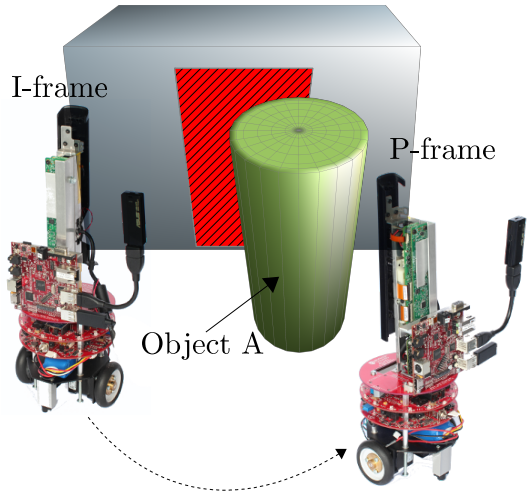


Figure 5.4: An example of situations that may lead to forward estimation failures: Object A is not in the field of view of the sensor at I-frame's location. The object enters the view of the sensor at P-frame's location, and a section of the background surface consequently becomes occluded.

can be described as

$$\begin{bmatrix} \frac{i_I - i_c}{f_x} & \frac{j_I - j_c}{f_y} & 1 & \frac{1}{z_I} \end{bmatrix}^T = \mathbf{M}^{-1} \begin{bmatrix} \frac{i_P - i_c}{f_x} & \frac{j_P - j_c}{f_y} & 1 & \frac{1}{z_P} \end{bmatrix}^T. \tag{5.4}$$

Pixels at (i_P, j_P) in $\mathbf{Z}_P$ can be mapped to (i_I, j_I) in $\hat{\mathbf{Z}}_I$. In this process, the pixels representing the range information of the green surface on object A will move out of the image coordinate range and will not be shown in $\hat{\mathbf{Z}}_I$. Therefore, we need to find the pixels in $\mathbf{Z}_P$ that move out of the image coordinate range in the reverse check process to complete the newly observed information determined by forward estimation.

C. Block-Based Update

In order to identify the newly observed depth information in the current depth frame, forward estimation and reverse check mechanisms are used in combination.

First, based on the transformation matrix $\mathbf{M}$, the forward estimation (Eq. 5.1) generates a virtual depth frame $\hat{\mathbf{Z}}_P$. Second, with the reverse check mechanism, $\hat{\mathbf{Z}}_I$ is generated from $\mathbf{Z}_P$ (Eq. 5.4). We record the coordinates of the pixels in $\mathbf{Z}_P$ which are warped out of the image coordinate range of $\hat{\mathbf{Z}}_I$. Third, we use these recorded

coordinates to set the values of the corresponding pixels in $\hat{\mathbf{Z}}_P$ to holes.

Then the virtual depth frame $\hat{\mathbf{Z}}_P$ and captured frame $\mathbf{Z}_P$ are uniformly subdivided into 8×8 pixels macro blocks. We search for the blocks in $\hat{\mathbf{Z}}_P$ with the number of pixels with invalid depth values which are above a pre-determined threshold[1]. Blocks in $\hat{\mathbf{Z}}_P$ can then be classified into two groups as follows,

if (number of pixels with invalid depth values in the block) $<$ threshold **then**

"*Skip block*": $\mathbf{Z}_I$ has sufficient data to predict the depth information in the block with the same coordinates in $\mathbf{Z}_P$.

else

"*Intra block*": The block with the same coordinates in frame $\mathbf{Z}_P$ contains the depth information that $\hat{\mathbf{Z}}_P$ does not have, and consequently cannot be predicted from $\mathbf{Z}_I$. The depth information in the block with the same coordinates in $\mathbf{Z}_P$ should be included in the encoding process.

end if

5.2.4 Differential Huffman Coding with Multiple Lookup Tables (DHC-M)

Using the process explained in Section 5.2.3, we can extract the newly observed depth information from every P-frame in a depth video. However, there is still further room for improvement. As the depth frames usually contain a large number of smooth regions, there is generally a high degree of correlation between adjacent pixels. Therefore, we can assert that a high degree of pixel-to-pixel correlation will result in a high compression ratio in differential coding. As the accurate information in keyframes [KM08] is a prerequisite for the successful operation of many applications [HKH+12], such as SLAM and 3D reconstruction, we need to keep the full fidelity of these frames which are usually the I-frames in a GoP.

[1]Based on different compression requirements, the value of the threshold can be set as a certain portion of the total number of pixels in one block, such as 1/2 or 1/6. A lower threshold leads to lower compression ratios.

Therefore, we propose a lossless coding scheme to encode complete I-frames and only the newly observed information in P-frames.

We designed our coding scheme, called *Differential Huffman Coding with Multiple Lookup Tables (DHC-M)*, to be fast and capable of compressing the depth images in a lossless manner without introducing any artificial refinements. The DHC-M scheme can be applied on complete depth frames or individual blocks.

The DHC-M scheme operates by comparing the current depth pixel with its reference pixel (which is the neighboring pixel of the pixel under consideration). If a pixel is at the end of a row, then it is treated as the reference pixel of the pixel in the same column under it (see Fig. 5.5).

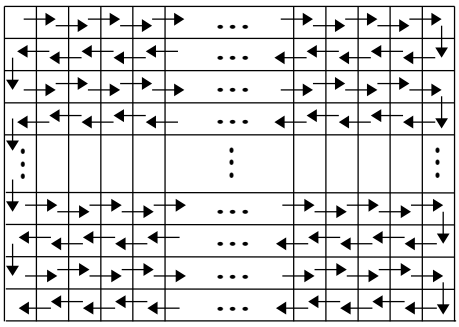


Figure 5.5: Depth pixels and their reference pixels in a frame.

The difference between the values of a pixel and its reference pixel is used in the encoding process. However, the depth image captured by RGB-D sensors has a proportion of pixels that have invalid depth value (holes). The invalid value is set at 2047, which is the largest number which can be represented by 11-bit data. This characteristic causes the difference between the current and reference pixels to have a high dynamic range. If we directly apply the standard entropy coding on the difference values, the compression ratio will be very limited. It is because the probabilities of difference values are distributed in a very wide range. In order to efficiently encode these difference values, we centralize the probability distribution by classifying the difference values into two groups according to the value of the reference pixel, and using two different lookup tables to encode the differences.

Let X_i represent the current pixel. If it has a valid depth value, we use the

symbol V_i, or H_i, if it is a hole. Similarly for its reference pixel, we use the symbols V_{i-1} or H_{i-1} depending on whether it is a hole or not.

The first lookup table is used for the cases when a reference pixel is a hole, and its values are generated according to the conditional probabilities as follows,

$$p(X_i|H_{i-1}) = \begin{cases} p(V_i|H_{i-1}) & \text{if the current pixel is valid} \\ p(H_i|H_{i-1}) & \text{otherwise,} \end{cases} \tag{5.5}$$

where $p(V_i|H_{i-1}) = q_h(V_i)$, and $p(H_i|H_{i-1}) = q_h(H_i - H_{i-1})$. The second lookup table is for the cases where the reference pixel has a valid depth value, and is established based on conditional probabilities as follows

$$p(X_i|V_{i-1}) = \begin{cases} p(V_i|V_{i-1}) & \text{if the current pixel is valid} \\ p(H_i|V_{i-1}) & \text{otherwise,} \end{cases} \tag{5.6}$$

where $p(V_i|V_{i-1}) = q_v(V_i - V_{i-1})$, and $p(H_i|V_{i-1}) = q_v(H_i)$. We use the standard technique of Huffman coding [Huf52] to generate two lookup tables based on the probability distributions of $q_h()$ and $q_v()$. Probability distributions of $q_h()$ and $q_v()$ are determined empirically by collecting information over a number of representative depth images which are captured in scenarios with varying geometrical structures.

In the encoding stage, the scheme first determines whether a reference pixel is a hole. If it is, the first lookup table (generated using Equation (5.5)) is used to determine the codeword of the difference value. Otherwise, the second lookup table (Equation (5.6)) is used instead.

5.2.5 Decoding Process and Under-Sampling Problem

At the decoder side, the received bitstream is decoded with the same lookup tables used at the encoder side. In the decoding process, if the reference pixel is a hole, the first lookup table is used to decode the current pixel, otherwise the second

table is used. As no motion compensation is applied on I-frames, every I-frame can decoded losslessly, while after motion compensation, only some blocks of each P-frame are transmitted with the transformation matrix. Therefore, each P-frame is lossy encoded and need to be decoded using its corresponding I-frame, transformation matrix, and transmitted blocks. We first apply the transformation matrix on each pixel in I-frame (using Equation (5.1)) to generate the prediction of the P-frame, $\hat{\mathbf{Z}}_P$. Then, we paste the transmitted blocks on $\hat{\mathbf{Z}}_P$. Eventually, the P-frame is reconstructed at the decoder side.

Directly applying warping equations may cause some visual artifacts in the synthesized view, such as disocclusions and cracks. Disocclusions are areas occluded in the reference viewpoint and which become visible in the virtual viewpoint, due to the parallax effect. Cracks are small disocclusions, mostly due to under-sampling issue. Some remarkable methods [ZT05, OYH09, ZL13, MFFT08] have been proposed to reduce these adverse effects, especially the effects of disocclusions. In our system, the disocclusion effects are fixed using information of the original P-frames in a block-based update process. Here, we only need to deal with the cracks generated due to the *under-sampling problem*. Under-sampling occurs when the warping process "rotates" a surface in such a way that the viewing angle becomes closer to the normal or reduces the distance between the sensor and the surface (Fig. 5.6). In such situations, the original image does not have enough information to predict the same surface from the viewpoint of the generated image.

A distinguishing feature of the pixels in cracks is that they have invalid depth values and their neighboring pixels usually have valid depth values. There are two general approaches to fill cracks. The first approach is to use the value of the nearest non-hole pixel on the left/right side to fill the pixels in the cracks [MMB97, LWP11]. This method is inaccurate and not robust for scenes that contain objects with complex geometries and variable camera motions. The second approach uses a median filter to smooth the whole image [MFFT08, OYH09]. This approach exhibits good performance on filling the cracks, but it smooths the complete image and

introduces noise in regions with correct depth values, especially on the object boundaries. In order to avoid this adverse effect, we have modified it by using an adaptive median filter. The filter is only applied on the pixels with invalid depth values instead of the whole image. A detailed performance evaluation of these three crack-filling algorithms is presented in Section 5.4.1.

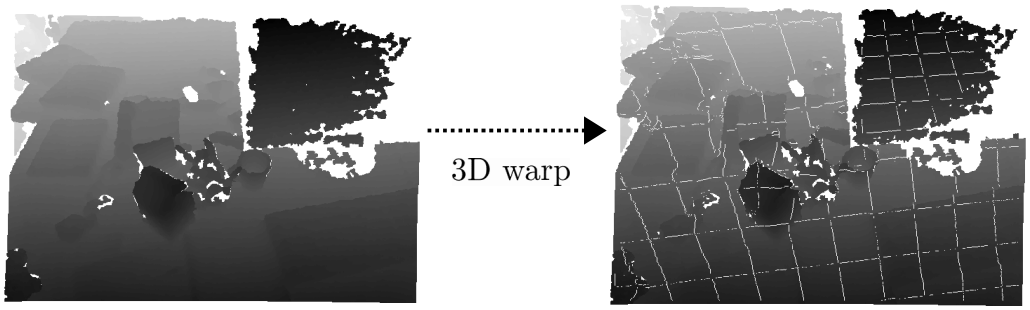


Figure 5.6: Cracks can be introduced during the image warping process due to the under-sampling problem.

5.3 Collaborative RGB-D Data Transmission for Multiple RGB-D Sensors

In the second circumstance, as the same scenario may be observed by multiple sensors, the collected images will inevitably contain a significant amount of correlated information, and the transmission load will be unnecessarily high if all the captured data are sent. In this section, we focus on this issue, and extend IW-DVC to a distributed framework to create a novel approach in developing a comprehensive solution for minimizing the transmission of redundant RGB-D data in VSNs. Our framework, called *Relative Pose based Redundancy Removal (RPRR)*, efficiently removes the redundant information captured by each sensor before transmission. We designed the RPRR framework particularly for RGB-D camera-equipped VSNs which eventually will require to work in situations under severely limited communication bandwidth. The scheme operates fully localized.

5.3.1 System Overview

In the RPRR framework, the characteristics of depth images, captured simultaneously with color data, are used to achieve the desired efficiency. Instead of using a centralized image registration technique [LCLL07, MSMW07], which requires one node to have the full knowledge of the images captured by the others to determine the correlations, we propose an approach based on relative pose estimation between pairs of RGB-D sensors [WŞD13a] and the 3D image warping technique [Feh04] to locally determine the color and depth information, which can only be seen by one sensor but not the others. Consequently, each sensor is required to transmit only the uncorrelated information to the remote station.

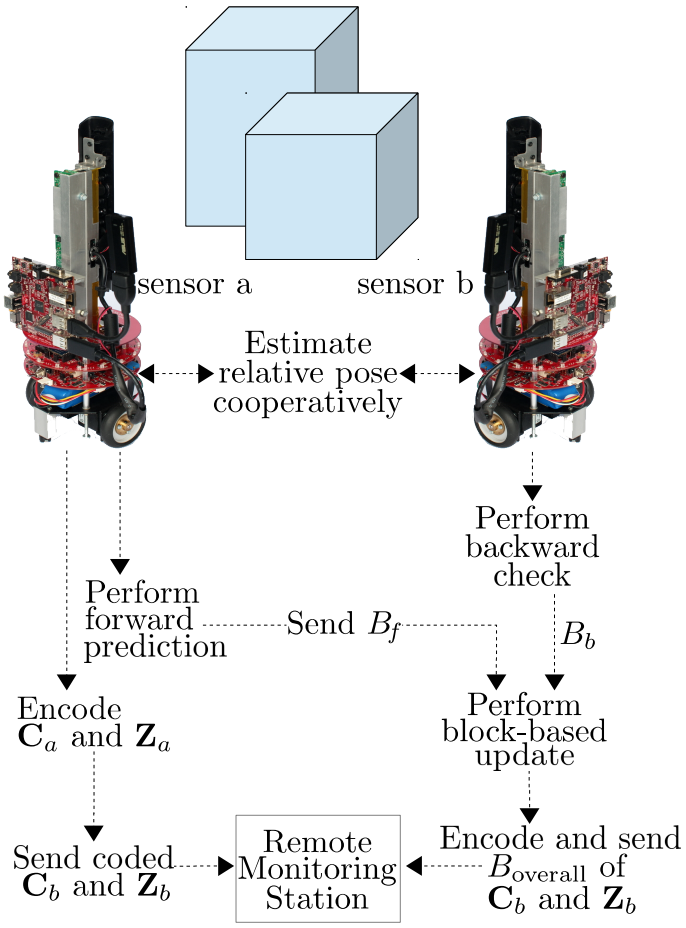


Figure 5.7: Operational overview of the RPRR framework.

Consider a simple sensor network consisting of two sensors. Let $\mathbf{Z}_a$ and $\mathbf{Z}_b$ denote a pair of depth images returned by Sensors a and b with overlapping FoVs. $\mathbf{C}_a$ and $\mathbf{C}_b$ are the corresponding color images. Before encoding depth and color

images to the bitstream, the disparities between the RGB-D information captured by the two sensors should be determined first. In the encoding procedure (see Fig. 5.7), we first estimate the relative location and orientation between two sensors. In the second step, forward prediction/backward check and block-based update using the relative pose information are performed to generate a prediction of $\mathbf{Z}_b$ and determine the depth information which is shown in $\mathbf{Z}_b$ but not shown in the $\mathbf{Z}_a$. Then, only the uncorrelated information in $\mathbf{Z}_b$ is encoded using a lossless entropy coding scheme. As the color image and depth image are registered, only the color information in $\mathbf{C}_b$ corresponding to the uncorrelated depth information needs to be transmitted. Therefore, the redundancy in the RGB-D information is removed in the encoding process.

In the decoding process, the received bitstream is decoded with the same coding algorithm used at the encoder side. Then, several post-processing approaches are proposed to deal with the under-sampling issue [MMB97] and enhance the quality of the reconstructed color and depth images.

To the best of our knowledge, the framework we present is the first distributed scheme that efficiently codes and transmits images captured by multiple visual sensors with large pose differences.

5.3.2 Relative Pose Estimation

As we require the relative pose information between two RGB-D sensors to determine the correlation in their captured images, the first step in this framework is relative pose estimation. The relative pose between RGB-D sensors a and b can be represented by a transformation matrix $\mathbf{M}_{ab}$ and can be computed by the relative pose estimation algorithm presented in Chapter 3.

5.3.3 Forward Prediction/Backward Check and Block-based Update

In this process, we use the same process described in Section 5.2.3 to determine the color and depth information of the regions in the scene which can be observed by both sensors. However, in this section, we distribute the process to two sensors.

In the forward prediction process, with the accurate relative pose information $\mathbf{M}_{ab}$, Sensor a can predict a depth image $\mathbf{Z}_b^*$, which is virtually captured at Sensor b's viewpoint, by applying Eq. 5.1 on each pixel in $\mathbf{Z}_a$. As the depth image is registered to the color image, the color pixels in $\mathbf{C}_a$ can also be mapped along with the depth pixels to generate a virtual color image $\mathbf{C}_b^*$. Then, all of the captured images and virtual images are decomposed into 8×8 macro blocks. In the virtual depth image, some blocks have no depth information, because none of the pixels in $\mathbf{Z}_a$ can be warped to these regions. This indicates the blocks with the same coordinates in $\mathbf{Z}_b$ and $\mathbf{C}_b$ contain the information that can only be observed by Sensor b but cannot be seen by Sensor a. Therefore, after Sensor a transmits these block coordinates to Sensor b, Sensor b will record these block coordinates as a set, B_f, and only needs to transmit the RGB-D information in these blocks of $\mathbf{Z}_b$ and $\mathbf{C}_b$ to the common receiver.

In order to comprehensively determine the redundancy, Sensor b also requires to generate virtual color and depth images captured by Sensor a. Similarly to the warping process from Sensor a to b, in the backward check process, Sensor b can also generate virtual images $\mathbf{Z}_a^*$ and $\mathbf{C}_a^*$, which are virtually captured at Sensor a's viewpoint. Thus Sensor b needs to determine the blocks including pixels in $\mathbf{Z}_b$ that move out of the image range in the backward check process. The set of these block coordinates is B_b. Then, Sensor b will derive the universe of the block coordinate sets B_f and B_b as $B_{overall} = B_f \cup B_b$. The blocks of $B_{overall}$ in $\mathbf{Z}_b$ and $\mathbf{C}_b$ contain the information which can only be observed by Sensor b.

In this process, each sensor can easily determine the uncorrelated RGB-D information by using only the relative pose information, and so avoid transmit-

ting/receiving and comparing complete color and depth images: Sensor a sends the complete captured color and depth images, Sensor b only sends the information in $B_{overall}$ to the remote monitoring station. As we show later in the experiment, this leads to significant bandwidth savings.

5.3.4 Image Coding

After the removal of the redundant information, the uncorrelated color/depth information is compressed to improve the efficiency of the communication channel usage. We use the coding scheme presented in Section 5.4.1 for encoding depth images. For RGB color data, we use the Progressive Graphics File (PGF) scheme developed by Stamm [Sta02].

There are many options for compressing color images, including JPEG2000 and H.264 intra mode. As the wireless channels are effected by noise and error prone, coding schemes that provide progressive coding are more suitable for sensor networks. Moreover, since a sensor node of a VSN has limited computational capability, a lightweight image coding scheme is required in sensor network applications. Progressive Graphics File (PGF) [Sta02], which is based on a discrete wavelet transform with progressive coding features, has high coding efficiency and low complexity. It has similar compression efficiency to JPEG2000, and 10 times faster than JPEG2000. Moreover, PGF has a small open source C++ codec [50] without any dependencies and is easy to use. Therefore, these properties make PGF suitable for on-board image compression, and we have implemented the PGF lossy mode in our testbed to compress two types of color information: color images and color image blocks.

5.3.5 Post-Processing at Decoder Side

At the decoder side, the received bitstream is decoded with the same lookup tables used at the encoder side. After the color and depth images captured by Sensor a are decoded, we use these to predict the depth and color images captured by Sensor b.

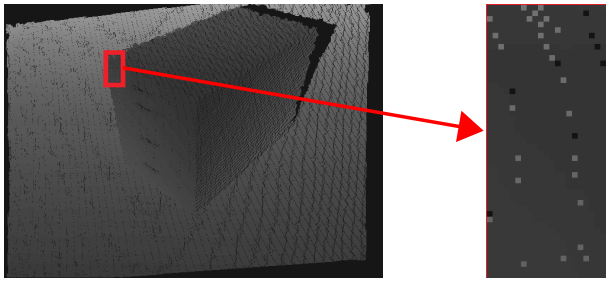


Figure 5.8: Ghost artifacts: The light gray pixels actually belong to the background surface and falsely warped onto the surface at the foreground.

As explained in Section 5.2.5, 3D image warping may introduce some visual artifacts in the synthesized view, especially when the transformation is large. In addition to the disocclusions and crack artifacts mentioned in Section 5.2.5, reconstructing images captured by Sensor b also introduces ghost artifacts due to the projection of pixels that have background depth and mixed foreground/background color.

In our framework, as the information that can only be observed by Sensor b is also transmitted, disocclusions can be eliminated by filling the disocclusion areas in the synthesized image with the color and depth information transmitted by Sensor b. Then, the main artifacts we need to deal with are cracks and ghosts. An intuitive example of the ghost artifact is shown in Fig. 5.8.

A. Removing Crack Artifacts

The missing color information in cracks is frequently avoided by operating a backward projection [MFFT08], which works in two steps. Firstly, the cracks in the synthetic depth image are filled by the median filter, and the bilateral filter is then applied to smoothen the depth map while preserving the edges. Secondly, the filtered depth image is warped back into the reference viewpoint to find the color of the synthetic view. This approach exhibits good performance in filling the cracks, but it smooths the complete image and introduces noise in regions with correct depth values, especially on the object boundaries. In order to avoid this adverse

effect, we have modified it by using an adaptive median filter to fill the cracks in reconstructed depth images. This filter is presented in Section 5.2.5. For the missing color information in cracks, instead of warping back the complete image to find the color information, we adopt the approaches of Do et al. [DZMdW09] which only warps back the filled depth pixels in cracks to find the corresponding pixels in the reference image, because the color information of the other pixels which are not in cracks can be directly estimated in the forward warping process.

B. Removing Ghost Artifacts

As illustrated in Fig. 5.8, some background surfaces are incorrectly shown on the foreground obstacle's surface, because the pixels representing the foreground surface become scattered after the warping process, and the background surface can be seen through the interspaces between these pixels. This artifact appears when the transformation/relative pose is very large. In order to remove this noise, we need to first identify the location of the incorrectly predicted pixels and then fill them with the estimated values. As the value of the incorrectly predicted pixel is significantly different from its neighboring pixels, this kind of impulse noise can also be revised by using the adaptive median filter. We propose a windowing scheme with 3×3 window size to determine whether or not a depth pixel contains an incorrect value. If more than half of the neighboring pixels are out of a certain range, being either much larger or much smaller than the centering pixel in the window, the centering pixel is estimated as a potential incorrectly predicted pixel. Then the centering pixel is replaced with the median value of its neighboring pixels which are out of the range. The corresponding color information can be found through backward warping which is similar to the solution for crack artifacts. *(We do not consider this artifact in Section 5.2, because the inter-frame motion is much smaller than the relative pose between two sensors. The foreground surface is not scattered enough to be seen through in the warping process at the decoder side of IW-DVC. Therefore, the ghost artifacts rarely appear in IW-DVC.)*

5.4 Performance Evaluation

In this section, we analyze the performance of the two frameworks, IW-DVC and RPRR.

5.4.1 Performance Evaluation of IW-DVC

We have implemented the IW-DVC framework in C++ using libCVD [lib], OpenCV [opea], and OpenKinect [opeb] libraries on a personal computer with an Intel i7-M620 2.66 GHz processor and 4GB memory. In order to evaluate its performance we have conducted a series of experiments using datasets selected from the archives provided by the Computer Vision Group at the Technical University of Munich [SEE+12]. All datasets in this archive consist of color and depth images captured by a mobile RGB-D sensor. The datasets also provide the intrinsic parameters of the RGB-D sensor used. All depth images were captured at a rate of 30 fps (full frame rate) with a resolution of 640×480 pixels. The pixels are trained to represent the distance in meters. For our experiments, we have selected the data sets containing images captured over static scenes with sufficiently rich geometrical features. The archival names of the selected datasets are included in Table 5.1. Before conducting the experiments, we have converted the depth images in these datasets back into their raw values so that each pixel has 2048 levels of sensitivity.

The two major components of the IW-DVC framework, motion compensation and the lossless coding scheme DHC-M have been tested separately. DHC-M has been tested compared against the JPEG2000 lossless mode and context-adaptive binary arithmetic coding (CABAC) [MSW03]. In the second set of experiments, the performance of the proposed motion compensation approach has been compared to the first motion compensation algorithm for depth video that uses 2D block-based motion vector sharing (2D-BMS) [GM04] and recently developed 3D block-based motion vector sharing approach (3D-BMS) [FWL11].

There are two main kinds of approaches to evaluate the quality of the recon-

structed depth images: (1) subjective, and (2) objective methods, such as peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Subjective methods [USS11, BHLCE13] require human viewers to train the assessment program and judge the quality of the reconstructed images. However, different from the conventional 8-bit depth images, the pixels in the depth images captured by RGB-D sensors contain 11-bit data. Original depth information in these images cannot be seen directly. If we downscale them to 8 bits per pixel before using subjective evaluation methods, a certain amount of information is inevitably lost. Therefore, the subjective methods are not applicable in this project, and consequently we have adopted objective methods to evaluate performances.

In the second set of experiments, we have used both PSNR and SSIM to evaluate the quality of the reconstructed depth images. We also have adopted the entropy approach to measure the differences between the reconstructed image and the uncompressed original image. Then, we have analyzed the performance of different crack-filling algorithms. In the third set of the experiments, we have analyzed the encoding complexity based on the processing time for each step and presented time complexity of each step in big O notation. For the latter three sets of experiments, the GoP size was set to 10 and the block size was 8×8.

A. Performance Evaluation of DHC-M Scheme

In the first set of experiments, in order to evaluate its effectiveness, we compared the DHC-M scheme against the standard JPEG2000 lossless mode and CABAC. Here, we have tested it alone without running the motion compensation block (Fig. 5.1).

DHC-M achieves its best performance when the individualized probability distributions are used to build lookup tables in different scenes. However, DHC-M cannot be universally used in this case. In this experiment, we randomly sampled 30 images in each dataset and generated lookup tables of DHC-M accordingly. So the same lookup tables can be used for all datasets, and DHC-M is able to achieve

its near-best performance. The results are presented in Table 5.1.

The size of a single uncompressed depth image is 614.4 KB. It is obvious that the DHC-M scheme outperforms JPEG2000 lossless mode and CABAC in all datasets. The average compression ratio of DHC-M is 18.7% higher than the average compression ratio of JPEG2000 and 127.9% higher than the average compression ratio of CABAC. It is because the images captured by RGB-D sensors have a much higher dynamic range than conventional gray-scale images and the pixels' values switch between the valid values and the invalid values (the largest value) frequently. DHC-M, which takes this feature into account and uses two lookup tables alternatively, centralizes the probability distribution of the difference values and achieves better context modeling. In contrast, JPEG2000 and CABAC, which are not specially designed for RGB-D sensors, do not consider this property. Furthermore, in lossless mode, JPEG2000 use a reversible wavelet transform and no quantization is performed. As a result, all bits planes have to be encoded, and JPEG2000 is not able to achieve the best performance.

B. Objective Evaluations of Reconstructed Depth Images

PSNR and SSIM can only indicate the degree of closeness between the reconstructed and original depth images. As the depth images captured by RGB-D sensors almost always have a number of pixels with invalid depth values, we need a quality evaluation method which can describe the noises of different kinds of pixels separately. Therefore, we use the entropy of the difference between the reconstructed depth image and the original depth image to evaluate the performance of the IW-DVC framework. As GoP size was 10, the system encoded one complete depth frame (I-frame) losslessly in every 10 frames. The redundancies in the other depth frames (P-frames) of the same group were removed by our motion compensation approach presented in Section 5.2.3. Then, only the newly observed depth information in these frames was coded by DHC-M.

Dataset		JPEG2000		CABAC		DHC-M	
Archive name	ID	Average Size (KB)	Comp. Ratio	Average Size (KB)	Comp. Ratio	Average Size (KB)	Comp. Ratio
freiburg1_plant	1	72.43	8.48	148.48	4.14	58.42	10.52
freiburg2_dishes	2	68.34	9.00	140.84	4.36	54.91	11.19
freiburg3_cabinet	3	60.02	10.24	118.72	5.18	51.40	11.95
freiburg3_large_cabinet	4	57.47	10.69	105.62	5.82	50.74	12.11
freiburg3_structure_texture_far	5	61.60	9.97	123.91	4.96	52.31	11.75
freiburg3_long_office_household	6	68.51	8.97	107.93	5.69	56.23	10.93
freiburg1_xyz	7	62.74	9.79	127.88	4.80	54.76	11.22

Table 5.1: Compression performance of JPEG2000, CABAC, and DHC-M.

Data set	Threshold	Comp. Ratio	PSNR	SSIM	Entropy		Average Number of Bits Required to Correct the Errors in a Reconstructed Depth Image (ANBR)		
					Pixels with Valid Value	Holes	Pixels with Valid Value	Holes	Overall
1	1/2	594	33.53	0.8356	1.49	1.07	3.46×10^5	8.03×10^4	4.36×10^5
	1/3	285	34.79	0.8585	1.45	0.81	3.38×10^5	6.28×10^4	4.00×10^5
	1/6	139	35.91	0.8818	1.44	0.65	3.34×10^5	4.78×10^4	3.82×10^5
2	1/2	590	35.66	0.8780	1.28	1.19	3.29×10^5	5.93×10^4	3.89×10^5
	1/3	374	37.88	0.8942	1.25	0.82	3.22×10^5	4.01×10^4	3.62×10^5
	1/6	256	38.71	0.9010	1.21	0.62	3.12×10^5	3.02×10^4	3.42×10^5
3	1/2	869	27.15	0.7254	1.47	1.92	3.79×10^5	9.53×10^4	4.74×10^5
	1/3	352	29.18	0.8144	1.45	1.75	3.73×10^5	8.73×10^4	4.60×10^5
	1/6	269	31.05	0.8321	1.40	1.53	3.60×10^5	7.73×10^4	4.38×10^5
4	1/2	395	42.87	0.9256	1.16	0.61	2.62×10^5	4.93×10^4	3.12×10^5
	1/3	268	43.26	0.9296	1.12	0.45	2.54×10^5	3.61×10^4	2.90×10^5
	1/6	211	43.88	0.9347	1.07	0.31	2.41×10^5	2.54×10^4	2.67×10^5
5	1/2	829	41.77	0.8770	1.13	0.61	2.99×10^5	2.59×10^4	3.25×10^5
	1/3	415	42.49	0.8883	1.11	0.55	2.93×10^5	2.34×10^4	3.16×10^5
	1/6	291	44.05	0.9040	1.09	0.48	2.88×10^5	2.26×10^4	3.09×10^5
6	1/2	625	46.51	0.9256	1.00	0.93	2.54×10^5	4.89×10^4	3.03×10^5
	1/3	342	48.06	0.9399	0.98	0.75	2.51×10^5	3.91×10^4	2.90×10^5
	1/6	220	49.26	0.9479	0.97	0.64	2.48×10^5	3.33×10^4	2.81×10^5
7	1/2	357	35.49	0.8117	1.25	1.45	2.86×10^5	1.14×10^4	4.01×10^5
	1/3	250	36.09	0.8303	1.20	1.26	2.75×10^5	9.96×10^4	3.74×10^5
	1/6	180	37.63	0.8592	1.13	0.96	2.58×10^5	7.49×10^4	3.33×10^5

Table 5.2: Quality evaluation of the reconstructed depth images generated by IW-DVC framework using different block update thresholds.

(1) System without Crack-Filling Algorithm

The objective of this experiment is to evaluate the effects of different block update thresholds on the compression ratio and the quality of the reconstructed depth image without implementing the crack-filling algorithm at the decoder side. Three thresholds were tested, which are 1/2, 1/3, and 1/6 of the overall number of pixels in one block. Two entropies for holes and pixels with valid values in the reconstructed depth images were derived. The entropies indicate the theoretical number of bits required to describe the difference between one pixel in the reconstructed depth frame and its corresponding pixel in the original uncompressed depth frame. We also derived the average number of bits required to correct the errors in the holes and pixels with valid values in one complete reconstructed depth frame to the original one (ANBR). The smaller ANBR, the higher the reconstructed quality achieved. The average entropies and ANBRs of seven datasets are presented in Table 5.2 with PSNR and SSIM.

According to Table 5.2, the compression ratio decreases when the threshold drops. This because more intra blocks need to be updated and encoded if the threshold is low. In the meantime, when more blocks in the original depth frame are updated, fewer prediction errors exist in the reconstructed depth image. Therefore, both entropies decline along with the threshold, and ANBR decreases simultaneously.

To verify the superiority of our system, we have implemented 2D-BMS [GM04] and 3D-BMS [FWL11] on DHC-M and tested them on the same datasets. 2D-BMS algorithm uses the motion vectors derived from the texture information to encode both color and depth video. 3D-BMS, developed recently, uses variable block sizes to perform motion estimation. In addition, it estimates motion vectors in z direction and good reconstruction quality on the depth video captured by a static camera has been reported [FWL11]. The results are presented in Tables 5.3 and 5.4.

By comparing the results in Tables 5.2, 5.4, and 5.3, it is clear that the 3D-BMS has enhanced accuracy than 2D-BMS, and our proposed motion compensation

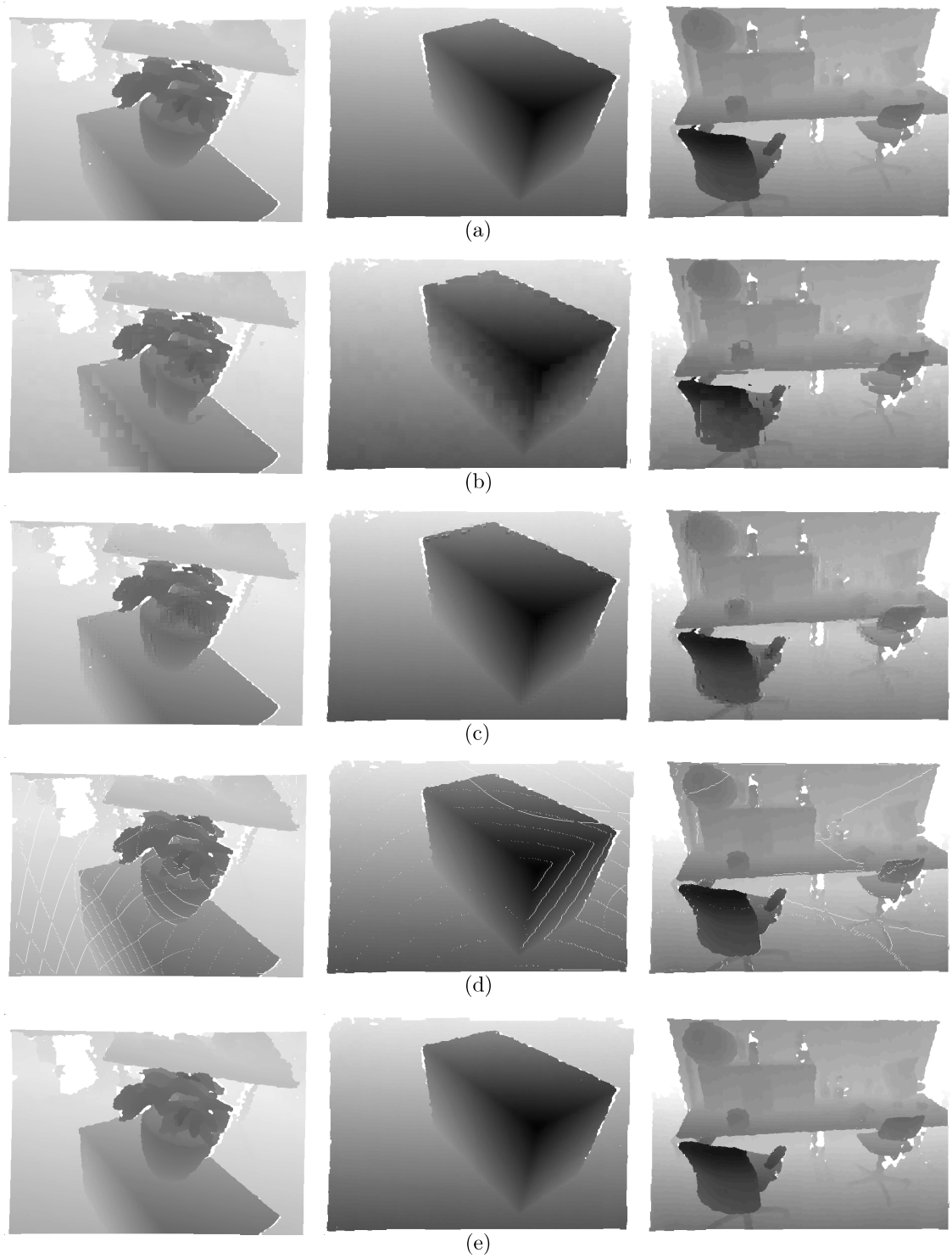


Figure 5.9: Examples of the original depth images and their reconstructed counterparts in datasets 1, 3 and 6: (a) Originals; (b) Reconstructed depth images generated by 2D-BMS; (c) Reconstructed depth images generated by 3D-BMS; (d) Reconstructed depth images generated by the IW-DVC framework with an update threshold of 1/3; (e) Enhancements of depth images in (d) by 3^{rd} crack-filling algorithm.

Data set	Comp. Ratio	PSNR	SSIM	Entropy		Avg. Number of Bits Required to Correct the Errors in a Reconstructed Depth Image (ANBR)		
				Pixels with Valid Value	Holes	Pixels with Valid Value	Holes	Overall
1	156	23.92	0.7321	2.82	1.84	6.54×10^5	1.39×10^5	7.93×10^5
2	265	24.27	0.8256	1.78	1.66	4.60×10^5	8.27×10^4	5.42×10^5
3	214	21.65	0.5471	1.95	1.46	5.04×10^5	7.13×10^4	5.75×10^5
4	220	35.76	0.8562	1.19	1.10	2.68×10^5	8.99×10^4	3.58×10^5
5	239	24.62	0.7707	2.73	0.77	7.33×10^5	2.95×10^4	7.62×10^5
6	189	42.21	0.9068	1.06	1.06	2.70×10^5	5.65×10^4	3.27×10^5
7	166	26.44	0.7759	2.38	1.36	5.53×10^5	1.02×10^5	6.55×10^5

Table 5.3: Quality evaluation of the reconstructed depth images generated by 2D-BMS.

Data set	Comp. Ratio	PSNR	SSIM	Entropy		Avg. Number of Bits Required to Correct the Errors in a Reconstructed Depth Image (ANBR)		
				Pixels with Valid Value	Holes	Pixels with Valid Value	Holes	Overall
1	118	28.68	0.7894	1.73	1.30	4.04×10^5	9.59×10^4	5.00×10^5
2	136	29.65	0.8730	1.40	0.95	3.64×10^5	4.67×10^4	4.11×10^5
3	149	27.06	0.7125	1.63	1.38	4.22×10^5	6.76×10^4	4.89×10^5
4	124	39.29	0.8723	1.05	1.04	2.37×10^5	8.54×10^4	3.22×10^5
5	125	33.62	0.8107	2.09	0.73	5.60×10^5	2.79×10^4	5.88×10^5
6	134	47.19	0.9184	0.99	0.93	2.51×10^5	5.06×10^4	3.01×10^5
7	115	32.18	0.8360	1.63	1.18	3.80×10^5	8.86×10^4	4.68×10^5

Table 5.4: Quality evaluation of the reconstructed depth images generated by 3D-BMS.

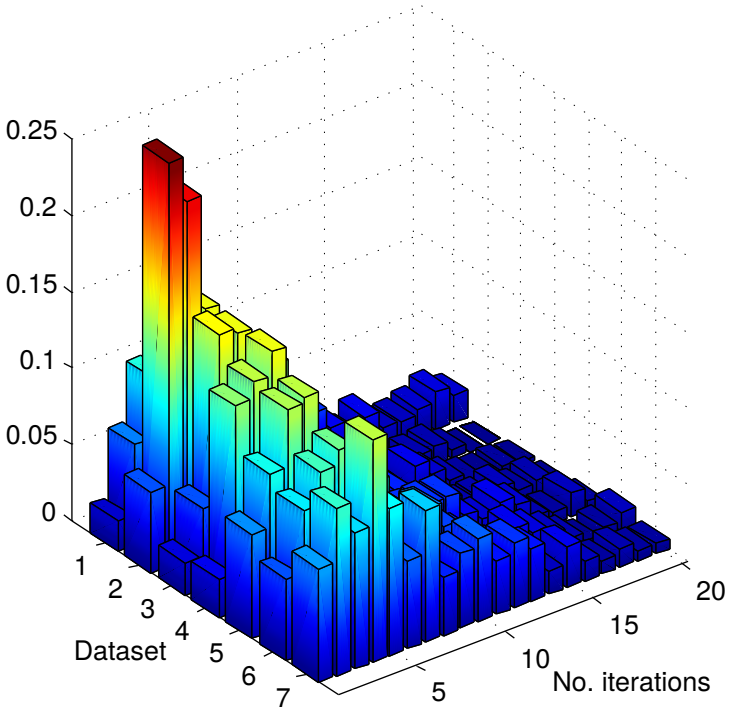


Figure 5.10: Probability distributions of the number of iterations required for each of the 7 datasets.

scheme with update threshold at 1/3 and 1/6 outperforms 2D-BMS and 3D-BMS in reconstruction accuracy, since in each dataset, our system has much smaller entropies and ANBR. The PSNR and SSIM results of our proposed methods are also higher than those for the other two algorithms. Moreover, our proposed method has higher compression ratios than 2D-BMS and 3D-BMS. One reason is that our approach needs only one 6D motion vector to describe the inter-frame motion. However, the conventional block-based motion compensation methods require a motion vector for each block.

We have selected three representative examples of results with noticeable differences/artifacts (Fig. 5.9). The scenarios in these three examples contain both smooth surfaces and abruptly changing object boundaries to allow us to check the performance of algorithms under a wide range of conditions. The smooth surfaces in the original images were reconstructed as coarse surfaces with block-artifacts by 2D-BMS (see the surface of the cabinet in the middle and left images of Fig. 5.9(b)). This is because the depth information of the same surface changes in the consecutive frames captured by a mobile RGB-D sensor, while 2D-BMS does

not consider this issue and does not change the values of the pixels representing the same surface in consecutive frames. 3D-BMS, which has an extra motion vector in z direction and uses variable block sizes, achieves relatively good performance in dealing with this problem. However, it still generates block artifacts on object boundaries, because different surfaces on the object boundaries can be placed in the same block. The distance changes on different surfaces are different. Then, the same motion vector in z direction is used for all the pixels in the same block, which leads to incorrect prediction on object boundaries. Therefore, we can see the block artifacts on the smooth surfaces are removed generally in the middle image of Fig. 5.9(c). But a large number of artifacts and blurs appear on the object boundaries in the left and right images of Fig. 5.9(c), when the scenes have complex geometrical structures. IW-DVC framework, which uses a 6D motion vector, not only maps each pixel from the reference frame to the predicted frame, but also changes its value individually. Therefore, the IW-DVC scheme does not introduce these block-artifacts and nicely preserves the object boundary information (see Fig. 5.9(d)). The results of this experiment are consistent with our analysis at the beginning of this paper that the block-based motion compensation schemes are suboptimal for depth video captured by a mobile sensor. However, in Fig. 5.9(d), it is obvious that some cracks regularly appear in the reconstructed depth images. This drawback of our motion compensation algorithm can be easily overcome by a crack-filling algorithm. This is addressed in the next section.

(2) System with Crack-Filling Algorithm

In this experiment, we have focused on determining the best performing crack-filling algorithm in recovering depth information of the cracks created due to the under-sampling problem. Based on the results of the experiments presented in Section 5.4.1, we have chosen 1/3 as the update threshold in this experiment by considering both compression ratio and the quality of the reconstructed depth images. We have tested the three crack-filling algorithms described in Section 5.2.5,

Table 5.5: Quality evaluation of the reconstructed depth images enhanced by different crack-filling algorithms. Update threshold = 1/3.

Data set	Method	Accuracy Percent	PSNR	SSIM	Entropy		Avg. Number of Bits Required to Correct the Errors in a Reconstructed Depth Image (ANBR)		
					Pixels with Valid Value	Holes	Pixels with Valid Value	Holes	Overall
1	1	48.7%	33.21	0.8645	1.62	0.04	3.95×10^5	2.48×10^4	3.97×10^5
	2	81.5%	35.77	0.8956	1.46	0.30	3.45×10^5	2.20×10^4	3.67×10^5
	3	81.5%	36.05	0.8985	1.45	0.23	3.44×10^5	1.67×10^4	$\underline{3.61\times10^5}$
2	1	20.2%	34.75	0.9286	1.30	0.05	3.43×10^5	2.10×10^3	3.45×10^5
	2	83.0%	39.41	0.9665	1.24	0.26	3.22×10^5	1.23×10^4	3.34×10^5
	3	83.0%	39.87	0.9676	1.25	0.19	3.25×10^5	9.08×10^3	$\underline{3.33\times10^5}$
3	1	26.6%	29.72	0.7167	1.65	0.09	4.41×10^5	3.64×10^4	4.44×10^5
	2	91.2%	32.67	0.8478	1.51	0.84	3.90×10^5	4.06×10^4	4.31×10^5
	3	91.2%	34.03	0.8574	1.46	0.34	3.79×10^5	1.66×10^4	$\underline{3.96\times10^5}$
4	1	18.8%	42.01	0.9520	1.17	0.06	2.77×10^5	4.07×10^3	2.82×10^5
	2	88.6%	46.27	0.9744	1.13	0.11	2.60×10^5	8.58×10^3	2.69×10^5
	3	88.6%	48.55	0.9776	1.13	0.07	2.59×10^5	5.47×10^3	$\underline{2.65\times10^5}$
5	1	32.5%	42.69	0.9179	1.17	0.05	3.18×10^5	1.92×10^3	3.20×10^5
	2	86.4%	43.77	0.9483	1.12	0.18	3.00×10^5	7.38×10^3	3.07×10^5
	3	86.4%	44.92	0.9506	1.12	0.14	2.99×10^5	5.72×10^3	$\underline{3.05\times10^5}$
6	1	42.4%	44.21	0.9184	1.05	0.06	2.75×10^5	2.60×10^3	2.78×10^5
	2	88.9%	46.99	0.9643	0.99	0.41	2.54×10^5	2.10×10^4	2.75×10^5
	3	88.9%	49.67	0.9679	0.99	0.26	2.54×10^5	1.33×10^4	$\underline{2.67\times10^5}$
7	1	31.9%	36.98	0.8611	1.32	0.16	3.22×10^5	1.07×10^4	3.33×10^5
	2	85.5%	37.50	0.8970	1.22	0.46	2.88×10^5	3.37×10^4	3.21×10^5
	3	85.5%	38.16	0.8993	1.21	0.41	2.84×10^5	3.00×10^4	$\underline{3.14\times10^5}$

and the results are presented in Table 5.5. The accuracy of each method has been calculated as follows

$$accuracy = \frac{correctly\_filled\_pixels}{\max(pixels\_need\_filling, filled\_pixels)} \times 100\%.$$

Tables 5.2 and 5.5 show that the third method has the highest accuracy, PSNR, and SSIM. The first method, which considers the horizontal neighboring pixels, is only suitable for cracks generated by a horizontally rotating and translating camera. However, in these datasets, the mobile camera moves and rotates in all directions. The second method has similar performance to the third. However, as it is applied on complete images, it smoothes the overall image and introduces noise in areas without cracks. Because of this, the entropy of holes decreases while the entropy of pixels with valid values increases. An important observation to make here is that the application of the third method leads to a significant reduction of prediction errors while maintaining a high compression ratio. The third method has higher ANBR, PSNR, and SSIM values, even though it uses an update threshold of 1/3 (Table 5.5) compared to 1/6 (Table 5.2). Therefore, we adopt the third crack-filling algorithm at the decoder side to enhance the quality of the reconstructed images and preserve high compression ratios. The corresponding enhanced results are shown in Fig. 5.9(e).

C. Encoding Complexity Analysis

In this set of experiments, we present time complexity of each step in big O notation and further analyzed encoding complexity based on the processing time of each step. We implemented the encoder with the update threshold at 1/3 on our computer. We measured the average processing time of each step required for encoding one frame in milliseconds. The results are summarized in Table 5.6.

In Fig. 5.10, we present seven histograms to illustrate the probability distributions of the numbers of iterations in different datasets. We conducted extra experiments and found there is not any direct relation between the number of sample points and the required number of iterations. We list the average number of

Table 5.6: Processing time of the proposed framework in each step for various datasets.

Dataset	Number of Iterations	Interframe Motion Estimation	Forward Estimation/ Reverse Check	Block Update	DHC-M	Overall
1	7.54	14.45 (31.12%)	26.95 (58.04%)	1.15 (2.48%)	3.88 (8.36%)	46.43
2	4.01	12.72 (30.22%)	24.10 (57.26%)	1.14 (2.71%)	4.13 (9.81%)	42.09
3	6.21	11.87 (27.29%)	25.51 (58.66%)	1.58 (3.63%)	4.53 (10.42%)	43.49
4	5.95	11.64 (28.16%)	24.82 (60.05%)	1.20 (2.90%)	3.67 (8.88%)	41.33
5	6.58	10.55 (24.11%)	27.98 (63.94%)	1.12 (2.56%)	4.11 (9.39%)	43.76
6	6.63	9.31 (23.32%)	25.26 (63.28%)	1.18 (2.96%)	4.17 (10.45%)	39.92
7	6.57	14.36 (33.43%)	23.38 (54.20%)	1.32 (3.07%)	3.99 (9.29%)	42.95
Average	6.21	*12.13 (28.31%)*	*25.41 (59.30%)*	*1.24 (2.89%)*	*4.07 (9.50%)*	*42.85*
Time complexity	—	$O(n_w n_s)$	$O(N_f)$	$O(N_f)$	$O(N_f)$	$O(N_f)$

iterations required for convergence in the process of inter-frame motion estimation in Table 5.6.

In Table 5.6, n_w and n_s represent the number of pixels in the searching window and the number of sampled points in the inter-frame motion estimation process. N_f represents the number of pixels in one complete frame. As $N_f \gg n_w n_s$, the time complexity of our motion compensation algorithm is $O(N_f)$. The time complexity of 2D-BMS is $O(N_w N_f)$. N_w indicates the number of different block positions in the matching window ($N_w \gg n_w$). The time complexity of 3D-BMS is also $O(N_w N_f)$. Without estimating MVs in a block-by-block manner, our proposed motion compensation approach with linear time complexity is more efficient than 2D-BMS and 3D-BMS. Moreover, according to Table 5.6, it is clear that the depth video can be encoded up to around 25 fps in real time on a standard computer.

5.4.2 Performance Evaluation of RPRR Framework

In this set of experiments, we evaluate the performance of the RPRR framework using two mobile RGB-D sensors of our VSN platform. Color and depth images were captured in six different scenes as shown in Fig. 5.11. In this setup, Sensor *a* transmits entire captured color and depth images to a central station (receiver). Then, Sensor *b* is required to only transmit the uncorrelated color and depth information that cannot be observed by Sensor *a* to the receiver. At the receiver, the color and depth images captured by Sensor *b* are reconstructed using the information transmitted by two sensors. As the entire color and depth images captured by Sensor *a* are compressed and transmitted to the receiver, we only have to evaluate the reconstruction quality of the images captured by Sensor *b*. The depth images are usually complementary to the color images in many applications, and in our framework the color images are reconstructed according to depth image warping. The reconstructed color images are necessarily related to the reconstructed depth images. If the color images are accurately reconstructed, the reconstructed depth images are also precise. Therefore, in this set of experiments we focused on

(a) Scene 1 (b) Scene 2 (c) Scene 3 (d) Scene 4 (e) Scene 5 (f) Scene 6

(g) Scene 1 (h) Scene 2 (i) Scene 3 (j) Scene 4 (k) Scene 5 (l) Scene 6

(m) Scene 1 (n) Scene 2 (o) Scene 3 (p) Scene 4 (q) Scene 5 (r) Scene 6

(s) Scene 1 (t) Scene 2 (u) Scene 3 (v) Scene 4 (w) Scene 5 (x) Scene 6

Figure 5.11: A demonstration of the scheme over six sets of images captured by the RGB-D sensors: First and second rows show the images captured by Sensors a, and b respectively. In the third row, image blocks transmitted by Sensor b are shown (here black regions denote the image blocks that are not transmitted). The fourth row shows the reconstructed images at the receiver side using the data sent by Sensor b.

evaluating the quality of the reconstructed color images.

A. Subjective Evaluation

The image blocks transmitted by Sensor b are shown in the third row of Fig. 5.11. The black regions in each image indicate the information which is not transmitted by Sensor b. The fourth row of the figure illustrates the reconstructed images using the data sent by Sensor b.

It can be seen that the images captured by Sensor a have been warped and stitched to generate the reconstructed color images captured by Sensor b. In the reconstructed images of scene 2 and scene 4, we also observe the significant color changes over the stitching boundary. This is because the illumination is inconsistent in the scene and the images captured by various sensors have different brightness. Generally, it is clear that the reconstructed images preserve the structural information of the original images precisely.

B. Objective Evaluation

Although many methods have been proposed to compress multi-view images [DVI$^+$12, CCAS12, CCM12, WC07, GD07, MMS$^+$09, LCLL07], they cannot be applied in our system, because these approaches either require the transmitter to have the knowledge of the full set of images or only work on cameras with very small relative pose. In our case, each sensor only has it own captured image and the relative pose between two visual sensors is very large. To the best of our knowledge, we propose the first distributed framework to efficiently code and transmit images captured by multiple visual sensors with large pose differences. We therefore do not have any work to compare ours against. For this reason, we can only compare the performance of our framework with the strategies which compress and transmit images independently.

As the color information is coded using PGF lossy mode, we can adjust the compression ratio, which leads to different coding performance. The performance was

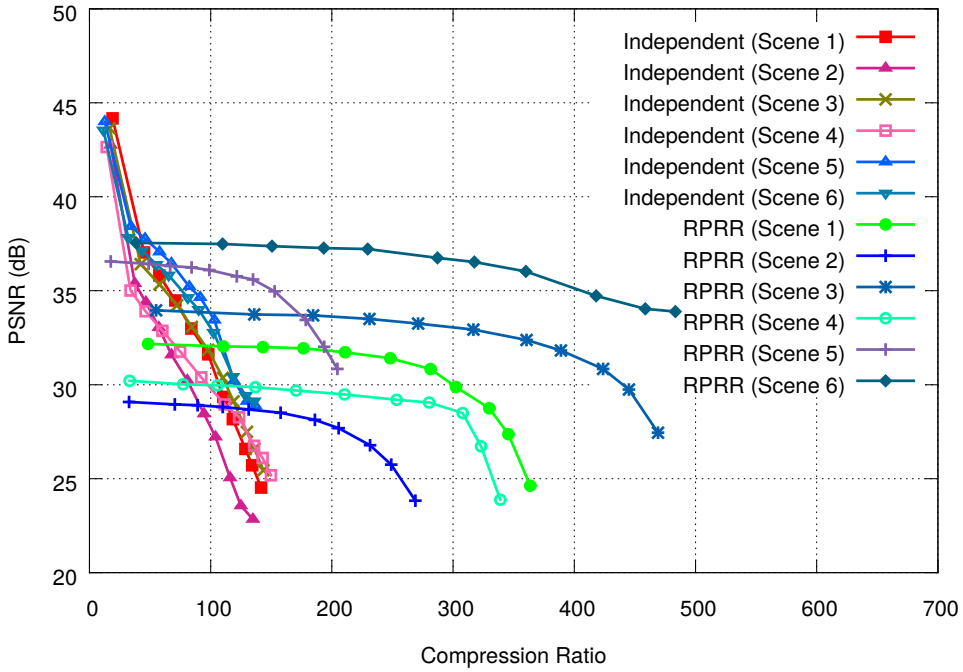


Figure 5.12: Comparisons of PSNR (dB) achieved by compressing the images at different compression ratios using the RPRR framework against transmitting them independently.

evaluated according to two aspects: reconstruction quality and compression ratio. We measured the Peak-Signal-to-Noise-Ratio (PSNR) between the reconstructed and original images captured by Sensor b at different compression ratios. The results are shown in Fig. 5.12. The line with square marks represents the result of collaborative coding and sending the uncorrelated image information, whereas the line with circle marks denotes the result of sending the entire image independently.

By referring to Fig.5.12, it is clear that RPRR framework can achieve much higher compression ratios than independent transmission scheme. However, the PSNR upper bounds achieved by RPRR framework are limited. It is because the reconstruction quality depends on the depth image accuracy and correlations between color images. Since the depth images generated by Kinect sensor is not accurate enough, the displacement distortion of depth images, especially the misalignment around the object edges, introduce noise in the reconstruction process. Another reason is the inconsistent illumination between the color images captured by two sensors. Even if the forward prediction/backward check process establishes the

correct correspondences between two color pixels according to the transformation between depth images, the values of these two color pixels can be very different due to the various brightness levels in two images. These characteristics lead to low PSNR upper bounds of the reconstructed color images. We can see that the reconstructed color image in Scene 6 has the highest PSNR, it is because the relative pose between two sensors is small, which leads to small differences in the structure of the captured scenes and the brightness of their captured images. Therefore, more information captured by sensor b can be reconstructed by information observed by sensor a. Therefore, according to Fig. 5.11 (f), only a small number of blocks in image captured by sensor b need to be transmitted. We also observe that Scene 2 and Scene 4 have the lowest reconstruction qualities, this is because the brightness level is quite different in the color images captured by two sensors (see image pairs shown in Fig. 5.11 (b)-(h), and Fig. 5.11 (d)-(j)). Although the structures of the scenes are preserved nicely in the reconstructed color images, distinct color changes over the stitching boundaries are shown in Fig. 5.11 (t) and (v). Subsequently, we can say that the RPRR framework is suitable to be implemented on the applications with very limited bandwidth which require very high compression ratios. It is because when the compression ratio increases, the quality of the color image reconstructed by RPRR decreases more slowly than the quality of the image compressed by the independent transmission scheme. Due to the large amount of captured color/depth data and limited bandwidth, our proposed RPRR framework fits well to the needs of VSNs equipped with RGB-D sensors.

C. Energy Consumption Evaluation

As their limited battery capacity on mobile sensors places limits on their performance, a data transmission scheme that minimizes the transmission load must not have a significant negative impact on the overall energy and bandwidth consumption. In this section, we present our comparative measurements of the overall energy and bandwidth consumption of the RPRR framework collected on our

eyeBug mobile visual sensors.

The overall energy consumption of the RPRR framework can be measured by

$$\begin{aligned} E^{R}_{\text{overall}} &= E_{\text{processing}} + E_{\text{encoding}} + E_{\text{sending}} \\ &= V_o I_p t_p + V_o I_e t_e + V_o I_s t_s \end{aligned} \tag{5.7}$$

in which V_o denotes the sensor's operating voltage, and I_p, I_e, and I_s represent the current drawn from the battery during processing, encoding, and sending operations. Also, t_p, t_e, and t_s are corresponding operation times required for these procedures.

The overall energy consumption when images are transmitted independently can be measured as,

$$\begin{aligned} E^{I}_{\text{overall}} &= E_{\text{encoding}} + E_{\text{sending}} \\ &= V_o I_e t_e + V_o I_s t_s. \end{aligned} \tag{5.8}$$

Note that, the operation times t_e and t_s are different in two transmission schemes as the image sizes change after removing the redundant information.

Our sensor operates at 15 V, and the current levels remain fairly constant during each operation. We measured them as follows: $I_p = 0.06$ A, $I_e = 0.06$ A, and $I_s = 0.12$ A. Our experiments show that in RPRR framework, due to different compression ratios, the transmission time varies between 32 and 42 ms, and the operational time for processing and encoding changes between 509 between 553 ms. The overall energy consumption of the RPRR scheme changes between 480 and 520 mJ depending on the compression ratio, and corresponding values for the independent scheme are between 918 and 920 mJ. The data clearly shows that the RPRR framework leads to the consumption of much lower battery capacity than the independent transmission scheme. It cuts the overall energy consumption of the sensor nearly by half. In RPRR framework, the energy consumption on two sensors are asymmetric, if sensor a always transmits complete images, the energy

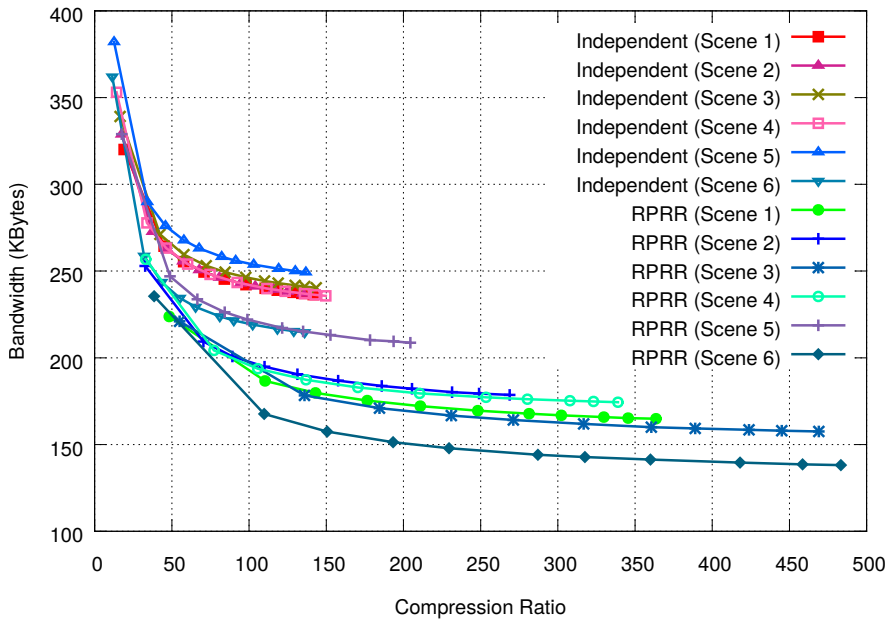


Figure 5.13: Comparisons of bandwidth consumption required at different color image compression ratios by using the RPRR framework against transmitting them independently.

will be quickly drained. A simple method to prolong the network lifetime is that two sensors transmit complete images alternatively. The current consumed by an eyeBug in idle status is 650 mA. According to the experimental results above, the theoretical operational time of RPRR on a pair of eyeBugs with 2500 mAh 3-cell (11.1 V) LiPo batteries is around 5.2 hours. In this period, around 3.24^4 color images with their corresponding depth images can be transmitted to the remote monitoring station.

Finally, we compare the overall bandwidth consumption on transmitting two depth images and two color images required by RPRR and independent transmission scheme. The results are presented in Fig. 5.13. We can see that since the compression ratio achieved by independent transmission scheme is limited, RPRR framework can achieve much lower the bandwidth consumptions than independent transmission scheme. It is also noticeable that when the compression ratio of the color image is the same, RPRR framework has a smaller bandwidth consumption. It is because only part of the color and depth images need to be transmitted in RPRR, while complete depth image has to be delivered in independent trans-

mission scheme. The data clearly shows that the RPRR framework leads to much lower bandwidth consumptions than the independent transmission scheme.

5.5 Concluding Remarks

In this chapter, we first present a novel coding framework, called 3D Image Warping based Depth Video Compression (IW-DVC), for efficiently removing the existing redundancy in depth video frames captured by a mobile RGB-D sensor. In particular, the motion compensation scheme included in the framework is designed to exploit the unique characteristics of depth images, and works cooperatively with the egomotion estimation and 3D image warping.

Experimental results show that our motion compensation method reduces the prediction errors of 2D-BMS and 3D-BMS for depth video captured by a mobile sensor to 55.9% and 72.8% on average, respectively. The result also demonstrate that the IW-DVC framework is capable of keeping the quality of the reconstructed depth image at a high level and can accurately determine the newly observed depth information in each frame. This significantly enhances the compression ratio. Furthermore, the results show that the IW-DVC framework is capable of operating in real time. As a final note, with the losslessly encoded I-frames, IW-DVC is suitable for many applications that have high requirements for the accuracy of keyframes.

In addition to the depth video coding scheme for a single sensor, we also introduce a novel collaborative transmission framework that efficiently removes the redundant visual information captured by the RGB-D camera-equipped nodes of a mobile VSN. We consider a multiview scenario in which pairs of sensors observe the same scene from different viewpoints. Taking advantage of the characteristics of depth images, our framework explores the correlation between the images captured by these sensors only using the relative pose information. Then, only the uncorrelated information is transmitted. This significantly reduces the

amount of information transmitted compared to sending two individual images independently. Experimental results show that the RPRR framework increases the compression ratio of the independent transmission scheme to 253.7% while it reduces the energy consumption by 54.7% on average. The RPRR framework is the first attempt to remove the redundancy in the color and depth information observed by VSNs equipped with RGB-D sensors. The system consists of only two mobile sensors at this stage. Redundancy removal between more than two sensors is related to optimally assigning sensors to different subgroups. This is another important topic in graph theory.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORKS

This thesis has dealt in the first place with sensor pose estimation in VSNs equipped with RGB-D sensors. Then, based on sensor pose information, the thesis has presented a depth video coding scheme and a complete collaborative image coding system architecture, yielding high-quality image rendering while retaining high compression efficiency, to achieve efficient transmission of color and depth information over bandwidth limited wireless channels. This chapter summarizes the achievements and points out several perspectives for sensor pose estimation and redundancy removal in VSNs which could be pursued as natural extensions of this work.

6.1 Summary of Contributions

The work presented in the preceding chapters has focused on providing new techniques for challenging problems in sensor pose estimation and efficient RGB-D information communication for VSNs, particularly in the GPS-denied environments and under severe communication constraints. The main contributions of this study can be summarized as follows:

Distributed Relative Pose Estimation Algorithm

Chapter 3 provides a novel method for 6DoF relative pose estimation between two RGB-D sensors, based on registration of the depth images captured by each

other. The proposed algorithm is able to operate in indoor environments which do not have GPS access. Our algorithm is based on the ICP algorithm, but explicitly accounts for the situation where two views of a scene each see parts that are occluded in the other view by making use of a beam imaging model implemented by reweighting the least squares operation in ICP. Further, the bias introduced by the beam model can be eliminated by symmetrizing across the two views. Finally, in order to make the algorithm practical, we distribute the working load of the algorithm to two sensors.

Self-Calibration Algorithm

Chapter 4 presents a new vision-based self-calibration algorithm for VSNs equipped with RGB-D sensors. We first model a VSN as an edge-weighted graph. Then, based on this model, and using real-time color and depth data, the sensors with shared FoVs estimate their relative poses in pairwise. The scheme does not require a single common view to be shared by all sensors, and it is able to operate in 3D scenes without any specific calibration patterns or landmarks. The proposed scheme evenly distributes working loads over the network. Therefore, the algorithm is scalable and the computing power of the participating sensors is efficiently used.

Depth Video Coding Scheme

In Chapter 5, we propose a new method, called *3D Image Warping Based Depth Video Compression (IW-DVC)*, for fast and efficient compression of depth images captured by mobile RGB-D sensors. We have designed the IW-DVC method to exploit the special properties of the depth data to achieve a high compression ratio while preserving the quality of the captured depth images. Our solution combines the sensor egomotion estimation and 3D image warping technique, and includes a lossless coding scheme which is capable of adapting to depth data with a high dynamic range. IW-DVC operates at high speed, is suitable for real-time applications, and is able to attain an enhanced motion compensation accuracy

compared with conventional approaches. It also removes the existing redundant information between the depth frames to further increase compression efficiency.

Collaborative RGB-D Data Communication

In Chapter 5, we present the *Relative Pose based Redundancy Removal (RPRR)* scheme for efficient color and depth information communication in bandwidth constrained operational scenarios. This scheme focuses on detecting and eliminating the transmission of redundant information gathered when multiple sensors have overlapping FoVs. Conventional approaches employ image comparison algorithms to determine the disparity among images and require at least one sensor to have the full knowledge of images captured by the others. These approaches are inevitably centralized, and are not able to eliminate the transmission of redundant information before its removal at each sensor. In contrast, in the RPRR scheme, the nodes determine their relative pose, and by using this knowledge they are able to distinguish the uncorrelated color and depth information from the rest locally. Consequently, this mechanism, which operates in a distributed manner, enables the nodes to transmit only non-redundant information. In this scheme, participating VSN nodes detect and remove the redundant visual and depth information, leading to a significant improvement in the efficiency of wireless channel usage.

With this study, we provide new tools to the sensor network and robotics communities, which will enable and foster sensor/robot localization and RGB-D information communication in VSNs, especially in bandwidth limited and GPS-denied scenarios.

6.2 Future Research Directions

In this section we propose some future research directions that would be interesting and useful to pursue.

The work on relative pose estimation in Chapter 3 is the first algorithm to

estimate the poses of multiple RGB-D sensors. This algorithm can also be used to register the depth frames captured by a single mobile sensor and derive the motion of the mobile sensor. However, our algorithm is constrained to operate in static scenes and it is not able to deal with dynamic objects in the environment. Therefore, future work will concentrate on estimating sensor pose in dynamic environments. There are very few studies focusing on this topic. The most straightforward idea is to distinguish dynamic regions from the scene in the captured depth images. As people walking around is the primary cause of the dynamic indoor scenarios, we can first identify the dynamic regions using pedestrian detection algorithms [EG09]. After the dynamic regions are determined, the algorithm operates only on the static regions of the captured depth frames. Thus, the adverse effects of the dynamic scenes can be removed. However, as the pedestrian detection algorithms can hardly operate in real time on computationally constrained sensors, the efficiency of the algorithm remains a challenging problem.

The self-calibration algorithm proposed in Chapter 4 is not fully distributed. In future, we plan to present a distributed solution to enhance the performance of the current semi-centralized framework. In order to achieve this goal, we require to upgrade the hardware of eyeBug. Thereby, eyeBug will obtain better computational ability and be able to operate more complicated computer vision algorithms. Further, we would like to develop an efficient algorithm which can keep updating sensors' relative poses in real time. This algorithm can be built on our framework to realize the refinement process in cooperative localization when sensors move in the scene.

The RPRR framework presented in Chapter 5 can only work on two RGB-D sensors at the moment. Another future research direction can be realizing efficient color and depth data communication in large RGB-D camera-equipped VSNs. Determining the correlation among the images captured by more than two sensors requires assigning sensors with overlapping FoVs to the same subgroups. As the self-calibration method presented in Chapter 4 has the ability to determine

neighboring sensors, this goal can be achieved by constructing an extended RPRR framework. The extended RPRR framework is a combination of the self-calibration method and the RPRR framework. In this framework, the primary sensor will be selected as the root and all the other sensors will be linked to the primary sensor to form a hierarchical tree structure. Each sensor will be assigned a rank (the root has the lowest rank). Then, the redundancy in the observed images observed by nodes with the same parents can be removed using the RPRR framework. By operating this process from the leaves to the root rank-by-rank, the redundancy in the images captured by the overall network can be removed.

In this thesis, we investigated efficient communication problems when sensors have overlapping FoVs and removed the redundancy in the captured images. Instead of removing the redundancy after capturing the correlated images, the third research direction can focus on preventing the redundant information from being captured. This involves camera selection and task assignment in VSNs considering the strong resource limitations. We consider VSNs in which each sensor can rotate its orientation. After determining sensor locations and orientations, the next thing we need to do is to rotate the orientations of sensors to provide the best possible coverage of events happening within the area of interest. In this process, the overlapping FoVs between sensors can be eliminated. In order to achieve this goal, greedy algorithms [SMGC08, XS07, LFSS11], evolutionary algorithms [DMR11, SDN$^+$12, SLKL13], and their combinations can be developed to maximize the coverage area while achieving optimal resource allocation in the network.

Bibliography

[ABC+03] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus. Tracking a Moving Object with a Binary Sensor Network. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 150–161, 2003.

[ABS08] C. T. Aslan, K. Bernardin, and R. Stiefelhagen. Automatic Calibration of Camera Networks Based on Local Motion Features. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, pages 16–22, 2008.

[ADB+04] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605 – 634, 2004. Military Communications Systems and Technologies.

[AH92] A. N. Akansu and R. A. Haddad. *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*. Academic Press, Inc., Orlando, FL, USA, 1992.

[AJ13] E. J. Almazan and G. A. Jones. Tracking People Across Multiple Non-Overlapping RGB-D Sensors. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2013)*, pages 831–837, June 2013.

[AZD13] D. S. Alexiadis, D. Zarpalas, and P. Daras. Real-Time, Full 3-D Reconstruction of Moving Foreground Objects From Multiple Consumer Depth Cameras. *IEEE Transactions on Multimedia*, 15(2):339–358, Feb 2013.

[BAA06] C. Bilen, A. Aksay, and G. B. Akar. A Multi-View Video Codec Based on H.264. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2006)*, pages 541–544, Atlanta, USA, 2006.

[BBD12] F. Bajramovic, M. Brückner, and J. Denzler. An Efficient Shortest Triangle Paths Algorithm Applied to Multi-Camera Self-Calibration. *Journal of Mathematical Imaging and Vision*, 43(2):89–102, 2012.

[BBD14] M. Brückner, F. Bajramovic, and J. Denzler. Intrinsic and Extrinsic Active Self-Calibration of Multi-Camera Systems. *Machine vision and applications*, 25(2):389–403, 2014.

[BCB+12] I. Barbosa, M. Cristani, A. Bue, L. Bazzani, and V. Murino. Re-identification with RGB-D Sensors. In *Proceedings of 2012 European Conference on Computer Vision Workshops (ECCVW 2012)*, volume 7583, pages 433–442. Springer Berlin Heidelberg, 2012.

[BD08] F. Bajramovic and J. Denzler. Global Uncertainty-based Selection of Relative Poses for Multi Camera Calibration. In *British Machine Vision Conference*, pages 1–10, 2008.

[BETVG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BHH11] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of Background Subtraction Techniques for Video Surveillance. In *Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 1937–1944, June 2011.

[BHLCE13] E. Bosc, P. Hanhart, P. Le Callet, and T. Ebrahimi. A Quality Assessment Protocol for Free-Viewpoint Video Sequences Synthesized from Decompressed Depth Data. In *2013 Fifth International Workshop on Quality of Multimedia Experience*, pages 100–105, July 2013.

[BM92] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[BSK+13] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions. In *Robotics: Science and Systems Conference (RSS)*, June 2013.

[CAS09] W. C. Chia, L. Ang, and K. P. Seng. Multiview Image Compression for Wireless Multimedia Sensor Network Using Image Stitching and SPIHT Coding with EZW Tree Structure. In *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2009)*, volume 2, pages 298–301, 2009.

[CCAS12] W. C. Chia, L. W. Chew, L. Ang, and K. P. Seng. Low Memory Image Stitching and Compression for WMSN Using Strip-based Processing. *International Journal on Sensor Networks*, 11(1):22–32, 2012.

[CCM12] S. Colonnese, F. Cuomo, and T. Melodia. Leveraging Multiview Video Coding in Clustered Multimedia Sensor Networks. In *Proceedings of 2012 IEEE Global Communications Conference (GLOBECOM 2012)*, pages 475–480, 2012.

[CDR07] Z. Cheng, D. Devarajan, and R. J. Radke. Determining Vision Graphs for Distributed Camera Networks Using Feature Digests. *EURASIP Journal on Applied Signal Processing*, 2007(1):220–220, 2007.

[CDS00] X. Chen, J. Davis, and P. Slusallek. Wide Area Camera Calibration Using Virtual Calibration Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 2, pages 520–527, 2000.

[CG10] Daniel G. Costa and Luiz Affonso Guedes. The Coverage Problem in Video-Based Wireless Sensor Networks: A Survey. *Sensors*, 10(9):8215–8247, 2010.

[CLF08] M. Calonder, V. Lepetit, and P. Fua. Keypoint signatures for fast learning and recognition. In *Proceedings of 2008 European Conference on Computer Vision (ECCV 2008)*, volume 5302, pages 58–71. 2008.

[CM91] Y. Chen and G. Medioni. Object Modeling by Registration of Multiple Range Images. In *Proceedings of 1991 IEEE International Conference on Robotics and Automation (ICRA 1991)*, pages 2724–2729, 1991.

[CMMV12] D. Chen, C. K. Mohan, K. G. Mehrotra, and P. K Varshney. Distributed In-Network Path Planning for Sensor Network Navigation in Dynamic Hazardous Environments. *Wireless Communications and Mobile Computing*, 12(8):739–754, 2012.

[CPS11] W. Choi, C. Pantofaru, and S. Savarese. Detecting and Tracking People Using an RGB-D Camera via Multiple Detector Fusion. In *Proceedings of 2011 IEEE International Conference on Computer Vision Workshops (ICCVW 2011)*, pages 1076–1083, Nov 2011.

[CSRL01] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[CSSH04] B. Chai, S. Sethuraman, H. S. Sawhney, and P. Hatrack. Depth Map Compression for Real-Time View-Based Rendering . *Pattern Recognition Letters*, 25(7):755–766, 2004.

[Cuc05] R. Cucchiara. Multimedia Surveillance Systems. In *Proceedings of the Third ACM International Workshop on Video Surveillance &Amp; Sensor Networks*, VSSN '05, pages 3–10, 2005.

[DH72] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[DJXay] I. Dryanovski, C. Jaramillo, and J. Xiao. Incremental Registration of RGB-D Images. In *Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA 2012)*, pages 1685–1690, May.

[DLL+11] N. D'Ademo, W. L. D. Lui, W. H. Li, Y. A. Şekercioğlu, and T. Drummond. eBug: An Open Robotics Platform for Teaching and Research. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2011)*, Melbourne, Australia, December 2011.

[DMKX12] I. Dryanovski, W. Morris, R. Kaushik, and J. Xiao. Real-Time Pose Estimation with RGB-D Camera. In *Proceedings of 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2012)*, pages 13–20, 2012.

[DMR11] B. Dieber, C. Micheloni, and B. Rinner. Resource-Aware Coverage and Task Assignment in Visual Sensor Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(10):1424–1437, Oct 2011.

[DR04] D. Devarajan and R. J Radke. Distributed Metric Calibration of Large Camera Networks. In *Proceedings of the First Workshop on Broadband Advanced Sensor Networks (BASENETS)*, volume 3, pages 5–24, 2004.

[DTPP09] I. Daribo, C. Tillier, and B. Pesquet-Popescu. Motion Vector Sharing and Bitrate Allocation for 3D Video-Plus-Depth Coding. *EURASIP Journal on Advances in Signal Processing*, pages 3:1–3:13, Jan 2009.

[DVI+12] N. Deligiannis, F. Verbist, A. C. Iossifides, J. Slowack, R. Van de Walle, R. Schelkens, and A. Muntenau. Wyner-Ziv Video Coding for Wireless Lightweight Multimedia Applications. *EURASIP Journal on Wireless Communications and Networking*, (1):1–20, 2012.

[DZMdW09] L. Do, S. Zinger, Y. Morvan, and P.H.N. de With. Quality Improving Techniques in DIBR for Free-Viewpoint Video. In *Proceedings of 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 1–4, May 2009.

[EG09] M. Enzweiler and D.M. Gavrila. Monocular Pedestrian Detection: Survey and Experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, Dec 2009.

[EHS+14] 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.

[EWK09] E. Ekmekçioğlu, S. T. Worrall, and A. M. Kondoz. A Temporal Subsampling Approach for Multiview Depth Map Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(8):1209–1213, 2009.

[eye11] eyeBug - a Simple, Modular and Cheap Open-Source Robot. `http://www.robaid.com/robotics/eyebug-a-simple-and-modular-cheap-open-source-robot.htm`, September 2011.

[FAT11] S. Foix, G. Alenya, and C. Torras. Lock-in Time-of-Flight (ToF) Cameras: A Survey. *IEEE Sensors Journal*, 11(9):1917–1926, 2011.

[FB81] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[Feh04] C. Fehn. Depth-Image-Based rendering (DIBR), Compression, and Transmission for a New Approach on 3D-TV. In *Electronic Imaging*, pages 93–104. International Society for Optics and Photonics, 2004.

[FSMA07] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth Mapping Using Projected Patterns, 2007.

[FSMA10] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth Mapping Using Projected Patterns. US Patent App. 2010/0118123 A1, 2010.

[FWL11] Y. Fan, S. Wu, and B. Lin. Three-Dimensional Depth Map Motion Estimation and Compensation for 3D Video Compression. *IEEE Transactions on Magnetics*, 47(3):691–695, March 2011.

[GD07] N. Gehrig and P. L. Dragotti. Distributed Compression of Multi-View Images Using a Geometrical Coding Approach. In *Proceedings of IEEE International Conference on Image Processing (ICIP 2007)*, volume 6, pages VI – 421–VI – 424, 2007.

[GIRL03] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the icp algorithm. In *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM 2003)*, pages 260–267, 2003.

[GM04] S. Grewatsch and E. Miiller. Sharing of Motion Vectors in 3D Video Coding. In *Proceedings of the International Conference on Image Processing (ICIP 2004)*, volume 5, pages 3271–3274, Singapore, 2004.

[Gru50] F. E. Grubbs. Sample Criteria for Testing Outlying Observations. *The Annals of Mathematical Statistics*, pages 27–58, 1950.

[GSB07] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[G09] P. Gomb. Detection of Interest Points on 3D Data: Extending the Harris Operator. In *Computer Recognition Systems 3*, volume 57, pages 103–111. Springer Berlin Heidelberg, 2009.

[HKH+12] 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. In *The International Journal of Robotics Research*, volume 31, pages 647–663, 2012.

[HL06] E. Hörster and R. Lienhart. Calibrating and Optimizing Poses of Visual Sensors in Distributed Platforms. *Multimedia systems*, 12(3):195–210, 2006.

[HS88] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, volume 15, page 50. Manchester, UK, 1988.

[HSXS13] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013.

[HT03] C. Huang and Y. Tseng. The Coverage Problem in a Wireless Sensor Network. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA 2003)*, pages 115–121, 2003.

[HTL12] W. Huang, C. Tsai, and H. Lin. Mobile robot localization using ceiling landmarks and images captured from an rgb-d camera. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2012)*, pages 855–860, July 2012.

[Huf52] D. A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, Sept 1952.

[HW77] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977.

[HWDK08] C. Hewage, S. T. Worrall, S. Dogan, and A. M. Kondoz. A Novel Frame Concealment Method for Depth Maps Using Corresponding Colour Motion Vectors. In *Proceedings of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 149–152, Istanbul, Turkey, 2008.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[IKH+11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of UIST*, pages 559–568, 2011.

[JSF12] H. Jaspers, B. Schauerte, and G. A. Fink. Sift-Based Camera Localization Using Reference Objects for Application in Multi-camera Environments and Robotics. In *Proceedings of International Conference on Pattern Recognition Applications and Methods (ICPRAM 2012)*, pages 330–336, 2012.

[KCTS01] R. Krishnamurthy, B. Chai, H. Tao, and S. Sethuraman. Compression and Transmission of Depth Maps for Image-Based Rendering. In *Proceedings of the International Conference on Image Processing (ICIP 2001)*, volume 3, pages 828–831, Thessaloniki, Greece, 2001.

[KFMK09] B. Kamolrat, W. A. C. Fernando, M. Mrak, and A. Kondoz. 3D Motion Estimation for Depth Image Coding in 3D Video Coding. *IEEE Transactions on Consumer Electronics*, 55(2):824–830, 2009.

[KGS05] P. Kulkarni, D. Ganesan, and P. Shenoy. The Case for Multi-Tier Camera Sensor Networks. In *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 141–146, 2005.

[KGSL05] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. SensEye: A Multi-tier Camera Sensor Network. In *Proceedings of the 13th Annual ACM International Conference on Multimedia (ACMM 2005)*, pages 229–238, 2005.

[Kho11] K. Khoshelham. Accuracy Analysis of Kinect Depth Data. In *Proceedings of ISPRS workshop on Laser Scanning*, volume 38, page W12, 2011.

[kin] KINECT camera. `http://www.xbox.com/en-US/kinect/default.html`.

[KK13] S. Kim and J. Kim. Occupancy Mapping and Surface Reconstruction Using Local Gaussian Processes with Kinect Sensors. *IEEE Transactions on Cybernetics*, 43(5):1335–1346, 2013.

[KKF12] M. Krainin, K. Konolige, and D. Fox. Exploiting segmentation for robust 3d object matching. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2012)*, pages 4399–4405, 2012.

[KLB08] G. Kurillo, Z. Li, and R. Bajcsy. Wide-Area External Multi-Camera Calibration Using Vision Graphs and Virtual Calibration Object. In *Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2008)*, pages 1–9, 2008.

[KM08] G. Klein and D. Murray. Improving the Agility of Keyframe-Based SLAM. In *Proceedings of European Conference on Computer Vision (ECCV 2008)*, volume 5303, pages 802–815. 2008.

[KQ11] M. Karakaya and H. Qi. Distributed Target Localization Using a Progressive Certainty Map in Visual Sensor Networks. *Ad Hoc Networks*, 9(4):576–590, 2011.

[KS11] J. Kelly and G. S. Sukhatme. Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[KSA+01] I. Kitahara, H. Saito, S. Akimichi, T. Ono, Y. Ohta, and T. Kanade. Large-scale virtualized reality. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages 1–4, 2001.

[KTCS09] L. Kneip, F. Tache, G. Caprari, and R. Siegwart. Characterization of the compact hokuyo urg-04lx 2d laser range scanner. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 1447–1454, 2009.

[LCH11] C. Lee, B. Choi, and Y. Ho. Efficient Multiview Depth Video Coding Using Depth Synthesis Prediction. *Optical Engineering*, 50(7):077004–077004–14, 2011.

[LCLL07] J. Lu, H. Cai, J. Lou, and J. Li. An Epipolar Geometry-Based Fast Disparity Estimation Algorithm for Multiview Image and Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, June 2007.

[LF06] T. Läbe and W. Förstner. Automatic Relative Orientation of Images. *Proceedings of the 5th Turkish-German Joint Geodetic Days*, 29:31–35, 2006.

[LFSS11] X. Li, H. Frey, N. Santoro, and I. Stojmenovic. Strictly Localized Sensor Self-Deployment for Optimal Focused Coverage. *IEEE Transactions on Mobile Computing*, 10(11):1520–1533, Nov 2011.

[lib] libCVD - Computer Vision Library. `http://www.edwardrosten.com/cvd/`.

[Low99] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV 1999)*, volume 2, pages 1150–1157, 1999.

[LTDL12] W. L. D. Lui, T. J. J. Tang, T. Drummond, and W. H. Li. Robust Egomotion Estimation Using ICP in Inverse Depth Coordinates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2012)*, pages 1671–1678, May 2012.

[LWP11] J. Lee, H. Wey, and D. Park. A Fast and Efficient Multi-View Depth Image Coding Method Based on Temporal and Inter-View Correlations of Texture Images. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(12):1859–1868, 2011.

[LXW+12] W. Liu, T. Xia, J. Wan, Y. Zhang, and J. Li. *RGB-D Based Multi-attribute People Search in Intelligent Visual Surveillance*, volume 7131, pages 750–760. 2012.

[Mar99] W. R. Mark. *Post-Rendering 3D Image Warping: Visibility, Reconstruction and Performance for Depth-Image Warping*. PhD thesis, University of North Carolina at Chapel Hill, 1999.

[MFFT08] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto. View Generation with 3D Warping Using Depth Information for FTV. In *Processings of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 229–232, May 2008.

[ML05] H. Ma and Y. Liu. Correlation Based Video Processing in Video Sensor Networks. In *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*, volume 2, pages 987–992, 2005.

[MMB97] W. R. Mark, L. McMillan, and G. Bishop. Post-Rendering 3D Warping. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 7–ff., Providence, USA, 1997.

[MMLB13] Z. Miljković, M. Mitić, M. Lazarević, and B. Babić. Neural Network Reinforcement Learning for Visual Control of Robot Manipulators. *Expert Systems with Applications*, 40(5):1721–1736, 2013.

[MMN13] A. Mogelmose, T.B. Moeslund, and K. Nasrollahi. Multimodal Person Re-Identification Using RGB-D Sensors and a Transient Identification Database. In *Proceedings of the International Workshop on Biometrics and Forensics (IWBF 2013)*, pages 1–4, 2013.

[MMS+09] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, H. N. de With Peter, and T. Wiegand. The Effects of Multiview Depth Video Compression on Multiview Rendering. *Signal Processing: Image Communication*, pages 73–88, 2009.

[MP11] M. Metzger and G. Polakow. A Survey on Applications of Agent Technology in Industrial Process Control. *IEEE Transactions on Industrial Informatics*, 7(4):570–581, 2011.

[MQC11] P. Morreale, F. Qi, and P. Croft. A Green Wireless Sensor Network for Environmental Monitoring and Risk Identification. *International Journal of Sensor Networks*, 10(1):73–82, 2011.

[MSMW07] P. Merkle, A. Smolic, K. Muller, and T. Wiegand. Efficient Prediction Structures for Multiview Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1461–1473, 2007.

[MSW03] D. Marpe, H. Schwarz, and T. Wiegand. Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):620–636, July 2003.

[NDI+11] 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 *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2011)*, pages 127–136, Basel, Switzerland, 2011.

[Nis04] D. Nister. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, June 2004.

[NMD13] V. Nguyen, D. Min, and M. N. Do. Efficient Techniques for Depth Video Compression Using Weighted Mode Filtering. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):189–202, 2013.

[opea] OpenCV: Open Source Computer Vision Library. `http://opencv.org`.

[opeb] OpenKinect Library. `http://openkinect.org`.

[OS12] M. F. Othman and K. Shazali. Wireless Sensor Network Applications: A Study in Environment Monitoring System. *Procedia Engineering*, 41:1204–1210, 2012.

[OYH09] K. Oh, S. Yea, and Y. Ho. Hole Filling Method Using Depth Based In-Painting for View Synthesis in Free Viewpoint Television and 3-D Video. In *Picture Coding Symposium*, pages 1–4, May 2009.

[OYVH09] K. Oh, S. Yea, A. Vetro, and Y. Ho. Depth Reconstruction Filter and Down/Up Sampling for Depth Coding in 3-D Video. *IEEE Signal Processing Letters*, 16(9):747–750, 2009.

[PCSM13] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, 34(3):133–148, April 2013.

[PHS87] A. Puri, H.-M. Hang, and D. L. Schilling. An Efficient Block-Matching Algorithm for Motion-Compensated Coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1987)*, volume 12, pages 1063–1066, 1987.

[Pic04] M. Piccardi. Background Subtraction Techniques: A Review. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2004)*, volume 4, pages 3099–3104, Oct 2004.

[PLT07] J. M. Phillips, R. Liu, and C. Tomasi. Outlier robust icp for minimizing fractional rmsd. In *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pages 427–434, 2007.

[PS03] S.-Y. Park and M. Subbarao. A fast point-to-tangent plane technique for multi-view registration. In *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM 2003)*, pages 276–283, 2003.

[PVFC13] A. Prati, R. Vezzani, M. Fornaciari, and R. Cucchiara. Intelligent Video Surveillance as a Service. In *Intelligent Multimedia Surveillance*, pages 1–16. 2013.

[RBB09] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proceeding of the IEEE International Conference Robotics and Automation (ICRA 2009)*, pages 3212–3217, May 2009.

[RD06a] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *Computer Vision ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. 2006.

[RD06b] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *Proceedings of the 9th European Conference on Computer Vision (ECCV 2006)*, volume 3951, pages 430–443. Graz, Austria, 2006.

[RHH08] V. Rodehorst, M. Heinrichs, and O. Hellwich. Evaluation of Relative Pose Estimation Methods for Multi-Camera Setups. *International Archives of Photogrammetry and Remote Sensing*, pages 135–140, 2008.

[RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings of the third International Conference on 3-D Digital Imaging and Modeling (3DDIM 2001)*, pages 145–152, 2001.

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2011)*, pages 2564–2571, 2011.

[RYMZ13] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, Aug 2013.

[Sch12] C. K. Schindhelm. Evaluating SLAM Approaches for Microsoft Kinect. In *Proceedings of the Eighth International Conference on Wireless and Mobile Communications (ICWMC 2012)*, pages 1691–1696, 2012.

[SDN+12] S. Sengupta, S. Das, M. Nasir, A. V. Vasilakos, and W. Pedrycz. An Evolutionary Multiobjective Sleep-Scheduling Scheme for Differentiated Coverage in Wireless Sensor Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6):1093–1102, Nov 2012.

[SEE+12] 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 (IROS 2012)*, pages 573–580, Oct 2012.

[SH09] S. Soro and W. Heinzelman. A Survey of Visual Sensor Networks. *Advances in Multimedia*, 2009, 2009.

[SHG02] T. Svoboda, H. Hug, and L. J. V. Gool. ViRoom - Low Cost Synchronized Multicamera System and Its Self-calibration. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 515–522, 2002.

[SHJH08] K. Shafique, A. Hakeem, O. Javed, and N. Haering. Self Calibrating Visual Sensor Networks. In *Proceedings of the IEEE Workshop on Applications of Computer Vision,*, pages 1–6, Jan 2008.

[SL04] S. Shih and J. Liu. A Novel Approach to 3-D Gaze Tracking Using Stereo Cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):234–245, 2004.

[SLKL13] J. Seok, J. Lee, W. Kim, and J. Lee. A Bipopulation-Based Evolutionary Algorithm for Solving Full Area Coverage Problems. *IEEE Sensors Journal*, 13(12):4796–4807, Dec 2013.

[SMAP14] S. Shahriyar, M. Murshed, M. Ali, and M. Paul. Inherently Edge-Preserving Depth-Map Coding Without Explicit Edge Detection and Approximation. In *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW 2014)*, pages 1–6, July 2014.

[SMFF07] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25:578–596, 2007.

[SMGC08] G. Simon, M. Molnar, L. Gonczy, and B. Cousin. Robust K-Coverage Algorithms for Sensor Networks. *IEEE Transactions on Instrumentation and Measurement*, 57(8):1741–1748, Aug 2008.

[SMW07] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.

[SPMP05] T. Svoboda, T. Pajdla, D. Martinec, and T. Pajdla. A Convenient Multi-Camera Self-Calibration for Virtual Environments, 2005.

[SSCZ13] J. Shen, P-C. Su, S. S. Cheung, and J. Zhao. Virtual Mirror Rendering With Stationary RGB-D Cameras and Stored 3-D Background. *IEEE Transactions on Image Processing*, 22(9):3433–3448, Sept 2013.

[Sta02] C. Stamm. A New Progressive File Format for Lossy and Lossless Image Compression. In *Proceedings of International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*,, pages 30–33, Czech Republic, 2002.

[TAT+12] Y. Takeda, N. Aoyama, T. Tanaami, S. Mizumi, and H. Kamata. Study on the indoor slam using kinect. In *Proceedings in Information and Communications Technology*, volume 4, pages 217–225. 2012.

[TBF05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[TM02] D. T and M. Marcellin, editors. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2002.

[TM08] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, (3):177–280, July 2008.

[Tsa87] R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

[TZL+12] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3D Full Human Bodies Using Kinects. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):643–650, 2012.

[USS11] A. S. Umar, R. M. Swash, and A. H. Sadka. Subjective quality assessment of 3d videos. In *Proceedings of AFRICON 2011*, pages 1–6, Sept 2011.

[VLMW08] J. Vergés-Llahı, D. Moldovan, and T. Wada. A New Reliability Measure for Essential Matrices Suitable in Multiple View Calibration. *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 114–121, 2008.

[VWS11] A. Vetro, T. Wiegand, and G.J. Sullivan. Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard. *Proceedings of the IEEE*, 99(4):626–642, 2011.

[WC07] M. Wu and C. W. Chen. Collaborative Image Coding and Transmission over Wireless Sensor Networks. *EURASIP Journal on Advanced Signal Processing*, 2007(1):223–223, Jan 2007.

[WHY11] H. Wang, C. Huang, and J. Yang. Block-Based Depth Maps Interpolation for Efficient Multiview Content Generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(12):1847–1858, 2011.

[WLC15] C. Wang, Z. Liu, and S. Chan. Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera. *IEEE Transactions on Multimedia,*, 17(1):29–39, 2015.

[WML+12] H. Wang, W. Mou, M. H. Ly, M. W. S. Lau, G. Seet, and D. Wang. Mobile robot ego motion estimation using ransac-based ceiling vision. In *PRoceedings of the 24th Chinese Control and Decision Conference (CCDC 2012)*, pages 1939–1943, 2012.

[WNB03] R. Wagner, R. Nowak, and R. Baraniuk. Distributed Image Compression for Sensor Networks Using Correspondence Analysis and Super-Resolution. In *Proceedings of the International Conference on Image Processing (ICIP 2003)*, volume 1, pages I – 597–600, Sept 2003.

[WPWS07] H. Wang, D. Peng, W. Wang, and H. Sharif. Optimal Rate-Based Image Transmissions via Multiple Paths in Wireless Sensor Network. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2007)*, pages 2146 –2149, July 2007.

[WSBL03] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H. 264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[WŞD+] X. Wang, Y. A. Şekercioğlu, T. Drummond, E. Natalizio, and I. Fantoni. Relative Pose Based Redundancy Removal: Collaborative RGB-D Data Transmission in a Mobile Visual Sensor Network. *IEEE Transactions on Multimedia*, (under review).

[WŞD13a] X. Wang, Y. A. Şekercioğlu, and T. Drummond. A Real-Time Distributed Relative Pose Estimation Algorithm for RGB-D Camera Equipped Visual Sensor Networks. In *Proceedings of the 7th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)*, pages 68–74, Palm Springs, USA, 2013.

[WŞD13b] X. Wang, Y. A. Şekercioğlu, and T. Drummond. Multiview Image Compression and Transmission Techniques in Wireless Multimedia Sensor Networks: A Survey. In *Proceedings of the 7th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)*, pages 258–265, Palm Springs, USA, 2013.

[WŞD13c] X. Wang, Y. A. Şekercioğlu, and T. Drummond. PhD forum: Efficient Communication Scheme for Mobile Visual Sensor Networks Equipped with RGB-D Cameras. In *Proceedings of the 7th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)*, pages 278–279, Oct 2013.

[WŞD14] X. Wang, Y. A. Şekercioğlu, and T. Drummond. Vision-Based Cooperative Pose Estimation for Localization in Multi-Robot Systems Equipped with RGB-D Cameras. *Robotics*, 4(1):1–22, 2014.

[WŞD15a] X. Wang, Y. A. Şekercioğlu, and T. Drummond. Self-Calibration in Visual Sensor Networks Equipped with RGB-D Cameras. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 2289–2293, April 2015.

[WŞD+15b] X. Wang, Y. A. Şekercioğlu, T. Drummond, E. Natalizio, I. Fantoni, and V. Fremont. Fast Depth Video Compression for Mobile RGB-D Sensors. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–14, 2015.

[wsr] Wireless Sensor and Robot Networks Laboratory (WSRNLab). `http://wsrnlab.ecse.monash.edu.au`.

[XS07] X. Xu and S. Sahni. Approximation Algorithms for Sensor Deployment. *IEEE Transactions on Computers*, 56(12):1681–1695, Dec 2007.

[YG11] Z. Yao and K. Gupta. Distributed Roadmaps for Robot Navigation in Sensor Networks. *IEEE Transactions on Robotics*, 27(5):997–1004, 2011.

[ZCWL12] Y. Zou, W. Chen, X. Wu, and Z. Liu. Indoor Localization and 3D Scene Reconstruction for Mobile Robots Using the Microsoft Kinect Sensor. In *Proceedings of the 10th IEEE International Conference on Industrial Informatics (INDIN 2012)*, pages 1182–1187, July 2012.

[Zha00] Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000.

[ZHL10] J. Zhang, M. M. Hannuksela, and H. Li. Joint Multiview Video Plus Depth Coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2010)*, pages 2865–2868, Hong Kong, China, 2010.

[ZKP13] B. Zeisl, K. Koser, and M. Pollefeys. Automatic Registration of RGB-D Scans via Salient Directions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2013)*, pages 2808–2815, 2013.

[ZL13] C. Zhu and S. Li. A New Perspective on Hole Generation and Filling in DIBR Based View Synthesis. In *Proceeding of 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 607–610, Oct 2013.

[ZT05] L. Zhang and W. J. Tam. Stereoscopic Image Generation Based on Depth Images for 3D TV. *IEEE Transactions on Broadcasting*, 51(2):191–199, June 2005.