January 2013

Full 3D Reconstruction From Multiple RGB-D Cameras

Owen Watson
University of South Florida, opw@mail.usf.edu

Full 3D Reconstruction From Multiple RGB-D Cameras

by

Owen Watson

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Co-Major Professor: Sudeep Sarkar, Ph.D.
Co-Major Professor: Dmitry Goldgof, Ph.D.
Rangachar Kasturi, Ph.D.

Date of Approval:
March 27, 2013

Keywords: Camera Alignment, Feature Matching, Stereo Vision, Camera Networks, Dense Depth

DEDICATION

To my family, friends, mentors, and culture

ACKNOWLEDGMENTS

Thank you to Dr. Sarkar, Dr. Goldgof, and Dr. Kasturi for your inspiration and mentorship. Special thanks to Mr. Bernard Batson for your support and insight. This research was supported by NSF Florida Georgia Louis Stokes Alliance for Minority Participation Award HRD #1139850.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis describes a novel procedure for achieving full 3D reconstruction from multiple RGB-D cameras configured such that the amount of overlap between views is low. Overlap is used to describe the portion of a scene that is common in a pair of views, and is considered low when at most 50% of the scene is common. Compatible systems are configured such that interpreting cameras as nodes and overlap as edges, a connected undirected graph can be constructed. The fundamental goal of the proposed procedure is to calibrate a given system of cameras. Calibration is the process of finding the transformation from each camera's point of view to the reconstructed scenes global coordinate system. The procedure focuses on maintaing the accuracy of reconstruction once the system is calibrated.

RGB-D cameras gained popularity from their ability to generate dense 3D images; however, individually these cameras can not provide full 3D images because of factors like occlusions from and a limited field of view. In order to successfully combine views there must exist common features that can be matched or prior heuristics pertaining to the environment that can be used to infer alignment. Intuitively, corresponding features exist in overlapping regions of views. Combining data from pairs of overlapping views would provide a more full 3D reconstructed scene. A calibrated system of cameras is susceptible to misalignment. Re-calibration of the entire system is expensive, and is unnecessary if only a small number of cameras became misaligned. Correcting misalignment is a much more practical approach for maintaing calibration accuracy over extended periods of time.

The presented procedure begins by identifying the necessary overlapping pairs of cameras for calibration. These pairs form a spanning tree in which overlap is maximized; this tree is referred to as the alignment tree. Each pair is aligned by a two-phase procedure that transforms the data from the coordinate system of the camera at a lower level in the alignment tree to that of the higher. The transformation between each pair is catalogued and used to reconstruction of incoming frames from the cameras. Once calibrated, cameras are assumed to be independent and their successive frames are compared to detect motion. The catalogued transformations are updated on instances that motion is detected essentially correcting misalignment.

At the end of the calibration process the reconstructed scene generated from the combined data would contain relative alignment accuracy throughout all regions. Using this proposed algorithm reconstruction accuracy of over 90% was achieved for systems calibrated with the angle between the cameras 45 degrees or more. Once calibrated the cameras can observe and reconstruct a scene on every frame. This is reliant on the assumption that the cameras will be fixed; however, in a practical sense this cannot be guaranteed. Systems maintained over 90% reconstruction accuracy during operation with induced misalignment. This procedure also maintained the reconstruction accuracy from calibration during execution for up to an hour.

The fundamental contribution of this work is the novel concept of using overlap as a means of expressing how a group of cameras are connected. Building a spanning tree representation of the given system of cameras provides a useful structure for uniquely expressing the relationship between the cameras. A calibration procedure that is effective with low overlapping views is also contributed. The final contribution is a procedure to maintain reconstruction accuracy overtime in a mostly static environment.

1 INTRODUCTION

1.1 Motivation and Goals

3D scene reconstruction is a classic problem in computer vision. In general, partial scenes are reconstructed from a limited number of views. To fully unlock the capabilities of modeling a scene in 3D the the data used for reconstruction should be representative of all regions of the scene. A point cloud generated by an RGB-D camera is the result of stereo combining the generated RGB image and IR image. Although depth information is recovered by these cameras, the resulting image is still a 2.5 D description of the actual scene. There may exist portions of the scene that are un-measurable due to occlusions from the view point. This motivated the usage of multiple RGB-D images to achieve a full 3D description of a scene. In general, techniques for 3D reconstruction using RGB-D images combine scans from a single moving camera. These techniques build a 3D model of the scene over time by aligning and concatenating successive scans. In contrast, the goal is to provide the reconstructed 3D scene observed from multiple views simultaneously.

The goal of this research is to continuously register data from multiple RGB-D cameras with a high degree of accuracy. The proposed algorithm is tailored for systems that contain pairs of low overlapping cameras. A system of cameras is considered calibrated when the error in reconstructing the scene is below a specified threshold. The algorithm should be independent of the operating environment, but the scope of this thesis only concentrates on static environments. The realization of a real time system would require more powerful hardware and is outside the scope of this research, however; a discussion on the feasibility is presented in the last chapter.

The reconstructed scene needs a global coordinate system to register the data generated from sparse cameras. The proposed algorithm presents a method to construct a spanning tree structure that represents the connectivity amongst the cameras. The transformation from any camera to the global coordinate system is achieved by tracing the path back to the root node. This allows error to be measured by accumulating the alignment error between pairs of cameras.

The desired generality of the calibration procedure negated any heuristic assumption that could be made about the operating environment. It is proposed that using a known object will provide enough useful features for successful calibration. Segmenting the object from each scene expresses the known features in the orientation of the view point. Using these features for alignment allows for consistent comparison of the aligned results. Using these features the two-phase alignment procedure should accurately converge yielding the best transformation between the pairs. The error in alignment between pairs is bounded by a threshold, which effectively makes the overall calibration relative throughout the system of cameras.

For this research it is assumed that invalidation a calibrated system can become misaligned overtime. Re-calibrating the system is an expensive and ineffective way to correct misalignment. Instead, approximating the transformation that would register the misaligned camera's points correctly in the global coordinate system is a more practical solution. This allows the algorithm to correct misalignment of cameras independently of each other without necessitating the re-calibration of the entire system.

1.2 Thesis Organization

This thesis is partitioned into six chapters. Chapter 1 is the introductory chapter. This chapter discusses the motivation for this research, overviews the goals, and outlines major milestones.

Chapter 2 presents a review of literature related to this research. This entails a discussion on multi view camera calibration, robust feature selection and matching, and maintaining camera calibration. Although the goals for multi view camera calibration expressed in literature may differ from that of this research, interesting correlations between the approaches can be discovered. Once these correlations are identified, then the differences imposed by the desired outcomes are accented. Additionally, procedures for effective selection of features and aligning images created by RGB-D cameras are discussed and compared with the goals for this research. This chapter wraps up with a discussion of continuous alignment schemes and identifies similarities and differences with the goals of this research.

Chapter 3 discusses the calibration procedure. This discussion details the process of selecting alignment pairs in a system, and obtaining a coherent representation of the sparse data. The formalism of error is explained within pairs, and for the system in its entirety. Each portion of the two-phase alignment process is explained. This entails the method for robust feature selection, finding common heading and orientation of points, and minimizing the error between corresponding images. Also this chapter discusses the optimization procedure used in finding the transformations that yield the best overall alignment between pairs throughout the system.

Chapter 4 explains the procedure of maintaining the calibration of the system. This method works by correcting for misalignment of each camera independently of the others in the system. The objective of this procedure is to align incoming images from cameras to the coordinate of the generating camera's calibrated location. An in depth discussion on the realignment procedure is provided.

Chapter 5 presents the method of validation used to evaluate the resulting calibration, and results from calibration using the algorithm described from this research. This entails a detailed description of the objectives of the experiments, and the method for measuring the accuracy of reconstruction. Results are presented from the end of the calibration procedure, from the system after operation for an hour, and from the system when realignment was computed.

Chapter 6 offers the conclusions derived from this research, to include the problems encountered and the steps followed to overcome each obstacle. The chapter suggests theories, based on current research results, as to the direction of future research.

1.3 Contributions

The concept of using overlap as a means of describing how a group of cameras are connected is novel. The method presented to create the alignment tree is a intuitive solution to expressing the relationship between cameras unambiguously. Moreover, the alignment tree identifies the configuration that has the greatest probable accuracy. Essentially, this is the best starting point for the rest of the procedure. The calibration process is effective against cameras with low overlapping views. This leads to the formulation of an error metric to measure the accuracy of aligning data from overlapping views. Lastly, an algorithm for merging the results of feature matching and ICP is evolved to reduce the effect of error aggregation of a long observation period. This is essential to the success of observing static environments with imperfect hardware. Collectively, these contributions serve the formalism of a strategy for full 3D reconstruction of low overlapping views on a per frame basis.

2 LITERATURE REVIEW

2.1 Introduction

This chapter presents a review of literature related to this research. Correlating ideas are identified, as well as, assumptions and implementation decisions that make these procedures incompatible with the goals of this research. This chapter also provides background theory that forms the foundation for formulating the algorithm developed by this research.

2.2 Multi-View Calibration of Point Clouds

At the lowest level, techniques for multi-view calibration construct point clouds by combining multiple 2D images using triangulation. This is the approach used to create point clouds from the RGB and IR images generated by RGB-D cameras. Although this is technically multi-view calibration, this work begins at a higher level where it is assumed that each camera's RGB and IR camera are calibrated. Also, the system is homogenous in the sense that all the cameras in the system are the same; this is important for comparison with works that used a single moving camera.

The most common approach to multi-view calibration of point clouds described in literature work with views that differ by a change in the observing location of a single camera. The stream of point clouds generated by the moving camera is calibrated by aligning and registering successive pairs of point clouds. Shahram Izadi et al. uses this approach in [7] to build an interactive model in two phases. The first phase uses the camera to acquire data for the model by being moved around the desired scene. Once this phase is done the camera is used to acquire data that interacts with the previously registered data. Inherently the assumption is that the amount of overlap between the successive views in the model building phase is substantial. Hao Du et al. describes a method in [6] that builds dynamic maps of the environment for data observed by a wandering robot; on each frame the newly observed data is calibrated and added to the growing model of the environment. Similarly, this approach also has an underlying assumption of large amounts of overlap between successive views.

The major drawback to the approaches discussed previously is that updating the reconstructed model can only happen in the viewable areas. Literature that focuses on wide baselines aim to align point clouds generated from largely varying views. The underlying assumption in these works is that the observation point of view is different but, there remains a set of features that can be calculated and matched in each view. This is apparent in approaches similar to that described by Michael Ying Yang et al. in [14] that work with wide baselines. This suggests that ability to match features between images is not dependent on the angle between views entirely. The assumption that increasing the angle between views decreases the overlap holds when the cameras are focused on the same point, or share the same origin. Inherently, these approaches assume that the amount of overlap between the views is substantial. It is important to note that relation between the angle separating views and the amount of overlap is not consistent. The concept of overlap as a metric is novel; hence the lack of reporting within results from literature. However, the amount of overlap between views can be inferred from the details presented in literature. Table 2.1 displays the interpreted amount of overlap between views from some popular literature. This assessment is intended to clarify the goals of this research with the achievements documented in literature.

Table 2.1: Inferred Amount of Overlap from Related Works

Authors	Title	Year	Overlap
Shahram Izadi et al.	KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera	2011	80-90%
Michael Ying Yang et al.	Robust Wide Baseline Scene Alignment Based on 3D Viewpoint Normalization	2010	50-70%
Hao Du et al.	Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera	2011	80-90%
Nicola Fioraio et al.	Realtime Visual and Point Cloud SLAM	2011	80-90%
Yiben Liu et al.	A Point-Cloud-Based Multiview Stereo Algorithm for Free-Viewpoint Video	2010	50-70%
Peter Henry et al.	RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments	2003	80-90%
Christoph Strecha et al.	Dense matching of multiple wide-baseline views	2010	50-70%
Richard Newcombe et al.	Live Dense Reconstruction with a Single Moving Camera	2010	80-90%

2.3 Robust Feature Selection and Matching

The quality of selected features directly impacts the quality of the reconstructed scene. When the ratio of corresponding features to the number of features is low, more iterations are required for the matching algorithm to converge and the possibility of divergence increases [8]. 3D feature detectors prove to be effective and more efficient for point clouds [18], and robust to large variations in the views [14]. Low amounts of overlap do not hinder the ability of feature detectors to detect features, instead, the ability for matching features is impaired heavily. This suggests that there needs to be an adequate amount of overlap for successful feature matching.

These algorithms prove successful when strong features are selected, however; they do not address how to identify strong features in a general sense because the needs differ from application to application. An intuitive solution would be inserting an alignment object to compute features in environments that lack these naturally. However, wider baselines can expose ambiguity in an object's geometrical description. Using a known set of features will stabilize the results and improve the quality of the initial alignment estimate because heuristics pertaining to the alignment object can be used to guide the alignment process. Matching features on a known rigid body model can effectively simulate the effects of distortion, while minimizing the effects of noise.

The challenge for describing 3D features is finding key points; generally, this is provided by a visual feature descriptor like SIFT or uniform sampling. Intuitively, a 3D feature can make effective usage of geometry. Rusu et al. created point feature histograms described in [12] based on this principle. Quality features are described by approximating the geometrical changes in a K-sized region about a key point. This algorithm does require normals to be computed at every point in order to compute the feature. This procedure is implemented in [13] and works well with data that may be complicated geometrically. Deterioration of the recovered surface will distort the computed surface normal, which can make the resulting alignment spurious [16]. This is a major concern because of the noise recovered in the point clouds.

Iterative Closest Point (ICP) is the commonly used algorithm for aligning RGB-D images; however, it is shown that this algorithm is most effective when an good initial alignment is presented. When successive frames are used, the initial alignment step can be ignored since overlaying the two images is a strong beginning point [26]. However, when the images are generated from two separate cameras, the initial alignment step becomes imperative for the rest of the alignment to succeed. It is important to note that ICP is effective with datasets that are to be overlaid instead of concatenated. This means that whatever set of features are being matched using ICP needs to be represented well in both sets; hence, the accuracy of the result is directly impacted by the amount of overlap between the views.

2.4 Scene Reconstruction

In general, related algorithms aim to create a 3D model of the observed scene that becomes more complete over time. Works similar to [25] can be categorized a single observing entity (camera, robot, etc.) which updates a global 3D model with the portion of the scene that is currently being observed. These applications keep stale data until fresh data is acquired. When a portion of the model needs to be updated with freshly observed data there is a resolving step implemented to resolve any inconsistencies. Works similar to [10] acquire images taken from different vantage points and updates the 3D model on the occasions that an incoming image adds to the model. In contrast, this work performs scene reconstruction on a per frame basis. This means that on every frame created, the entire scene is reconstructed with all updates happening simultaneously.

The use of an global coordinate system which is independent of the camera's suggest that each cameras data should be aligned to this space individually [21]. Paton et al. outlines the concern of the accumulation of residual error incurred from the approximation of the aligning algorithms in [15]. This accumulation is noticed in long chains of alignments. However, cameras in this work fall into small clusters of overlapping groups which makes this concern negligible. It is also noted that if there is a camera that cannot be aligned then aligning a chain of cameras will fail. In this work each camera is required to overlap with at least one other camera in the system; hence, there is a guarantee that all cameras can be aligned pairwise.

The goal of this research is to perform per frame scene reconstruction. Since all viewable regions of the scene are updated together, there is no need to maintain a model of the scene. Each camera is constrained to overlapping with at least one other camera; hence, there is a guarantee that there is an complete alignment of the system. Also, it is assumed that there may be small clusters of cameras that all overlap with each other, but the error accumulated within these clusters is small enough to be negligible.

2.5 Maintaining Calibration

There are many algorithms for maintaining calibration of a system of cameras over time. Generally, re-calibrating the entire system is expensive and may require human intervention. Common practice is to recompute some constant features, or set of invariant features on every frame and update the calibration by computing the transformation between matching pairs. Dang et al. uses epipolar geometry as a constraint for locating and matching features then re calibrates a pair of cameras in [22]. It is important to note that features may not be in abundance, or located in enough regions of the scene to be considered a solid solution. Approaches maintain a 3D model and compare incoming points to existing points in order to correct alignment. This idea of correcting misalignment instead of recalibrating is adopted in this research.

Collecting multiple views and reconstructing a scene by updating a 3D model becomes cumbersome as changes from one camera may cancel out changes from another. This works well with approaches like [6] since updates only come from one source. Intuitively, the effect of correcting misalignment is best achieved by assessing each camera independently. Camera tracking is effective at registering incoming data accurately in relation to the position of the camera [7]. The major concern is the effect of accumulating error over the duration of observation.

Dang et al. shows that the error due to accumulation can be circumvented by combining the results from multiple alignment schemes. An alignment scheme based solely on features is threatened by the fact that there is no guarantee that features will always be recovered at the same location. Similarly, an alignment procedure based on ICP is threatened by the fact that there is noise in the recovered data, so matching is not an exact one to one. For this reason the approach in [22] combined feature matching and ICP in such a way that the accumulating error is minimized over long video sequences.This process makes a calibrated system robust and feasible to be incorporated into real world applications.

3 ACHIEVING CALIBRATION

3.1 Introduction

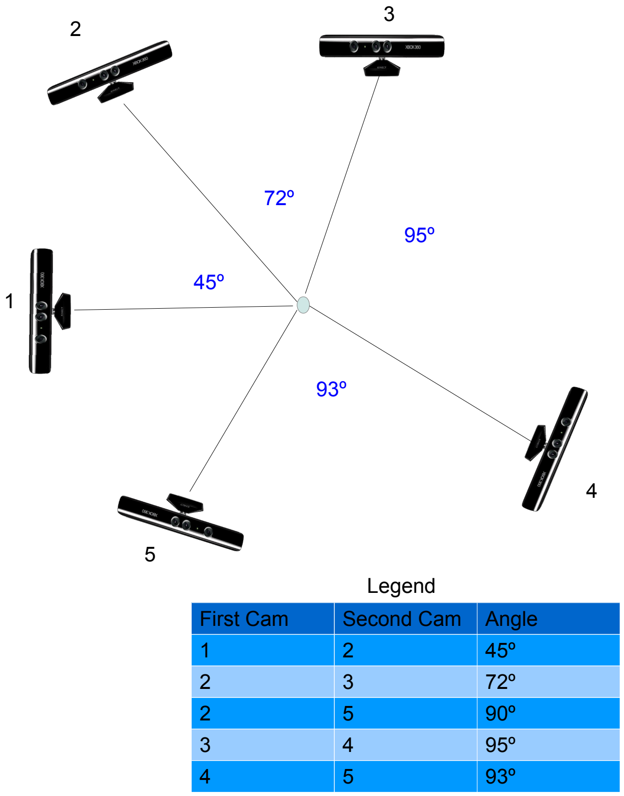
The RGB-D cameras used for this research were Microsoft Kinects. Each camera generated sequences of point clouds instead of separate color and depth images. The only restriction for the configuration of the cameras was each camera being required to have between 20% and 50% overlap with at least one other camera in the system. Otherwise, the proximity of the cameras can be arbitrary. For the alignment process a planar object with a unique geometric description is used to describe each scene. This object will be referred to as the alignment object. The alignment object needs to be distinctive enough to maintain a concise representation of its features across views. Each pair is presented the alignment object in a region that allows complete visibility in both cameras' field of view. As a result, the effects of the large change in perspective on rigid bodies can be estimated. This is useful for systems constructed in locations that may not contain enough descriptive information naturally in the environment for successful calibration.

Initialization begins with the construction of the alignment tree. This tree is the minimum spanning tree and connected nodes are interpreted as a pair. Each identified pair then undergoes a two-phase alignment process. The goal of the first phase is to identify the common heading and orientation of the two scenes and provide a rough alignment of the scenes. The second phase refines the alignment by minimizing the distance between corresponding points. The full calibration of the camera system is achieved by tracing the alignment tree from the furthest child nodes back to the root. The points from a child node are transformed and then concatenated with the parent's effectively transforming all clouds into one coordinate frame. The end result is the reconstructed scene.

3.2 Building the Alignment Tree

The first step in calibrating the system is the manual estimation of the amount of overlap between potential pairs of cameras. The amount of overlap was determined by a human observing the data from each pair of cameras. Larger camera angles suggest smaller amounts of overlap depending on the orientation of the cameras. Since cameras are not restricted to overlapping only one other camera there may be multiple ways to represent the connectivity structure of the system. In a probabilistic sense the highest potential accuracy for alignment exist between images that are closer together. This is the fundamental idea behind constructing the alignment tree as a minimum spanning tree.

To initialize the algorithm, a primary camera is arbitrarily chosen from the pair of cameras added to the alignment tree first. This camera's coordinate system is also assumed to be the global coordinate system for the reconstructed scene. Prim's algorithm for constructing the minimum spanning tree is created using the angle between the cameras as the criterion [23]. Conversely, when the amount of overlap and the angle between cameras is inconsistent the maximum spanning tree is created by selecting the camera with most overlap on each iteration. The resulting tree, referred to as the alignment tree, identifies the necessary connection between pairs of cameras needed for complete calibration of the system of cameras. Alignment pairs consist of a parent, and a child node. These pairs have the greatest potential of yielding accurate results because they will contain the greatest number of correspondences. This process is displayed in Figure 3.1 - Figure 3.3.



Legend

First Cam	Second Cam	Angle
1	2	45º
2	3	72º
2	5	90º
3	4	95º
4	5	93º

Figure 3.1: Example Configuration of a System of Cameras. Each camera is pointed towards the same point to simplify the example. It is also assumed that larger angles between cameras yield lower amounts of overlap. Solid black lines in the diagram denote the direction a camera is looking. The legend details the amount of overlap between pairs.

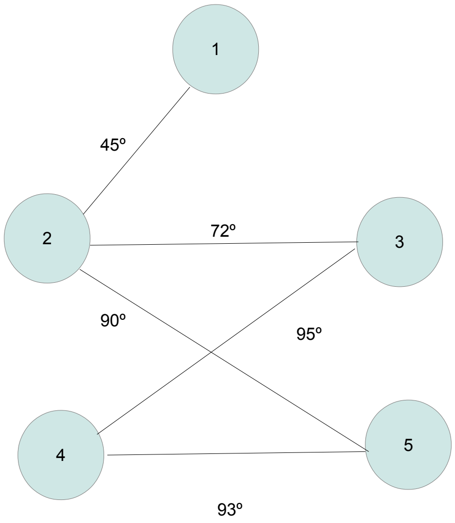


Figure 3.2: Example System of Cameras Interpreted as an Undirected Graph. The example system displayed in Figure 3.1 is converted into an undirected graph. The nodes represent cameras and the edges represent the amount of overlap between cameras.

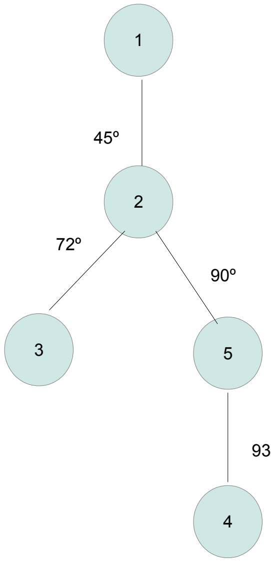


Figure 3.3: Alignment Tree Representation of the Example System of Cameras. The undirected graph displayed in Figure 3.2 is simplified to the spanning tree that maximizes the overlap through the system of camaeras.

The cloud corresponding to the i^{th} pair (p_i) of cameras at time t_j is defined as

$$p_i(t_j) = Source(t_j) \cup Transformed(Child(t_j))$$

where $Transformed(Child(t_j))$ is computed using the following formula.

$$\begin{bmatrix} Transformed(Child(t_j))_{x_i} \\ Transformed(Child(t_j))_{y_i} \\ Transformed(Child(t_j))_{z_i} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} Child(t_j)_{x_i} \\ Child(t_j)_{y_i} \\ Child(t_j)_{z_i} \\ 1 \end{bmatrix}$$

Once this is computed for every pair of cameras, any camera's points can be expressed in terms of the unified coordinate system by simply tracing the path back to the root and applying the transformations along the way. This is implemented by applying transformations a parent node endures to all of its children. Intuitively, a frame for the system would consist of the combined data from all cameras at a given time. This is defined for the i^{th} frame as

$$f_i = \bigcup_{j=1}^{j=n} p_j(t_i).$$

Each frame is interpreted as the union of all aligned pairs at a given time. The perspective of the reconstructed frame is that of the primary camera's. Conveniently, a frame can be translated to the perspective of any camera by applying the transformations that represent the path in the alignment tree from the primary to target camera.

3.3 Initial Alignment

The goal of the initial alignment phase is to find the corresponding heading and orientation of the cameras in a pair. Without any information about the physical location of the cameras, the data obtained from the images taken must be used to find this. When the amount of overlap between views decreases, the amount of variance between the features recovered and the difficulty of matching corresponding features increases. For stability in results an alignment object is inserted into the scene supplementing the need for a robust set of features. An asymmetric planar object is used to minimize the complications posed from features created from out of plane rotations, and reduce the spuriousness of alignment results due to surface degradation. An asymmetric shape has only one possible registration regardless of the angle, and ambiguity is minimized in cases of partial overlap. The segmented alignment object from each frame is used to model the data and select features from each perspective. Ideally, features used for alignment would be distributed throughout each image. For this research that is not the case since overlap is low, features are required to be in a tighter proximity to fit within the region of overlap.

Figure 3.4: Planar Object Used for Alignment. This asymmetrical planar object is used to supplement features in views for computation of the aligning transformation.

Identifying strong features is a well studied computer vision problem [1]. The first step is to finding features is identifying key points. SIFT features are calculated on the segmented alignment object and are in turn projected into 3D and used as key points on the alignment object. PFH features are preferred since they describe the local geometry of a k-sized region about a point of interest. Intuitively FPFH features require the surface normals at every point to be computed. RANSAC is implemented at a coarse level to find a rough alignment given the descriptive information for each scene. The major benefit to comparisons occurring only between points on the object is the increase in the quality of the initial alignment due to the increased possibility of finding relevant matches in the small area that is overlapped in the images.

This method is used for initial alignment because the results will be biased to the location of the features which may not be the optimal representation of the data. At convergence the only solid assumption is that the resulting alignment places points from each image in the general vicinity of the corresponding points. Since the data is concentrated in the area of overlap the best approximation would yield the least error for all corresponding points. For this reason there needs to be a refining phase that considers all the data available.

3.4 Refining Alignment

The goal of the refining phase of alignment is to find the transformation that most accurately aligns the (pre-aligned) images from the initial alignment phase. Iterative closest point(ICP) is used for refinement because of its proven effectiveness at registering 3D points; however, best results are achieved when the points start out roughly aligned [9]. ICP iteratively calculates and applies transformations that reduce the distance between corresponding points until a convergence criterion is met. At convergence, the returned transformation provides the best fit in a least square sense. The quality of the result is measured by the error between the aligned clouds. ICP is applied on a fine scale since corresponding points are close. The error between the aligned clouds of the i^{th} pair of cameras is defined as the summed euclidean distance of corresponding points

$$\epsilon(p_i) = \sum_{j=1}^{j=n} |Source(t_j) - Transformed(Child(t_j))|.$$

The definition of error within the i^{th} frame (f_i) is the total error between matching points of all pairs

$$E(f_i) = \sum_{j=1}^{j=n} \epsilon(p_j).$$

It becomes apparent that to minimize the error in a frame, the error between pairs of cameras must be minimized. The default implementation of ICP is prone to becoming stuck at local minima; moreover, there is no evaluation of the model-data fitting error. Considering that the depth data received from the RGB-D cameras is extremely noisy and larger widths in baselines have more dramatic deformation effects, this exposes a prime location for optimization. The Levenberg-Marquardt algorithm is a non-linear function optimizer [2]. When combined with ICP (LM-ICP), the selected transformation during each iteration minimizes both the distance between points, and the model-data fitting error [4]. This optimization also benefits from the prevention of converging to a local minima. The effectiveness of this optimization is complemented by the unique geometric representation of the alignment object. Generally, LM-ICP converges to a particular confidence in less iterations than the general ICP, although each iteration takes more time [1]. Theoretically ICP will converge after an infinite number of iterations; however, practical implementations bound the number of iterations by a specified parameter. Intuitively, the optimal transformation needs to be selected on each iteration for the best results.

The result at convergence of LM-ICP is considered to provide the minimal error in alignment between the pair of clouds. This is represented by

$$\epsilon_{min} = min(\sum_{j=1}^{j=n} |Source(t_j) - Transformed(Child(t_j))|)$$

The minimal error of the reconstructed frame can be approximated by

$$E_{min}(f) \approx \sum_{i=1}^{i=n} \epsilon_{min}(p_i).$$

Since each pair is restricted to converge with a less than the allowed error between pairs (ϵ_{min}), the quality in alignment amongst pairs is relatively similar assuring the calibration is consistent throughout the system.

4 MAINTAINING CALIBRATION

4.1 Introduction

During operation of the system each camera's stream is considered to be independent. The transformation from each camera to the unified coordinate system is known at the beginning of operation by tracing the alignment tree. The goal of this portion of the algorithm is to update the transformation for a misaligned camera and maintain the accuracy of the originally calibrated system. The major assumption is that cameras capture frames much faster than motion occurs; hence, the smallest amount of variance happens between subsequent frames. This yields high accuracy and fast convergence of the underlying alignment algorithm.

4.2 Identifying Scene Features

For this phase of the algorithm PFH features were also used for descriptors; however, the method for generating them differed from that outlined in the initial alignment section. Uniform sampling replaced SIFT features as the method for identifying key points to avoid having to determine the best scale to compute features. With the alignment object a set of scales can be derived through heuristics generated about the object. In contrast, this phase requires features to be generated from the observed data in the scene since the system will be in operation. Once the key points are found then the same method of calculating PFH features is implemented.

In this phase successive frames from a single camera are being aligned, which negates the assumptions of overlap presented earlier. Because of the large amounts of overlap features can be distributed about the image increasing the potential accuracy of the calculated alignment. Although a camera may observe a static scene from frame to frame, there is no guarantee that the scene will be recovered exactly the same. This makes the usage of feature matching infeasible as a stand alone solution. Intuitively, corresponding features are forced to be within a small radius of each other. This may potentially identify correct matches, but does not address the issue of aligning to features that are recovered in slightly different locations. For this reason feature matching is used in the manner of [15] with some variations outlined in the next section.

4.3 Computing Re-Alignment

Camera tracking is a classic approach to detecting motion. When used in conjunction with 3D reconstruction the primary concern becomes accumulative error due to the continuous calculation of the update to the transformation of a points data. Dang et al. shows in [22] that combining the results from multiple alignment algorithms greatly increases the overall accuracy of alignment. Although ICP is powerful, it is still plagued with the possibility of converging to a local minima, and was designed for non noisy data. It is also suggested to use ICP as a refinement to an approximation.

4.3.1 Feature Matching

In this phase it is crucial to identify good correspondences between features. For this reason, correspondences are required to be within a small distance from each other. When a set of strong features and correspondences are identified, a rigid body transformation between the two sets of features are calculated. At this point the transformation is concerned to be a rough alignment between the two frames. However, if a good set of correspondences can not be found then the transforming matrix from this phase is assumed to be the identity matrix. This assumption is valid when motion happens much slower than frames are created.

The work described in [22] simply uses the found alignment as input to ICP as an initial estimate. However, ICP treats the points of the input data the same. This only makes partial usage of the results from the feature matching phase when there are strong correspondences. When there is not any supplementing information obtained from the feature matching phase then running ICP alone is adequate. Intuitively corresponding features have their closest point identified. From this ICP should find solutions that keep correspondences close and minimize the fitting error of the other points. The derived process is discussed in the following section.

4.3.2 Combining Feature Matching Results with ICP

The ICP algorithm was designed to work with datasets without noise. Reasonable results are achieved with noisy data and optimizations are presented to help improve the results. Typically an initial alignment is provided to guide the algorithm to convergence. The previous phase of feature matching gives two options to the implemented ICP. On one hand feature matching failed and ICP is given the identity matrix as an initial alignment. In this case the best approximation is found by finding the best general fit from a fine scaled alignment process. On the other hand a transformation is provided that was calculated from the rigid body transformation of corresponding features.

Although this initial alignment hints the ICP algorithm to the best direction for optimal results, iterations can easily negate the results from the previous algorithm. Instead, the portion of the ICP algorithm that calculates the distance between points weighs those points that correspond to features more than other points in order to be better in line with the results from the previous phase. This forces ICP to address feature matches as closest points and makes the algorithm select updating transformations in which corresponding features are closest and all other points have the best fit.

5 EXPERIMENTS AND RESULTS

5.1 Introduction

The experiments performed aim to evaluate the accuracy obtained by systems of cameras calibrated using the proposed algorithm. The first set of experiments concentrate on evaluating the resulting precision from the calibration phase of the algorithm. The successive experiments focus on testing the ability for the algorithm to sustain the initial accuracy. The final set of experiments test the ability to maintain accuracy in a volatile environment.

5.2 Description of Calibration Experiments

For the following experiments camera systems containing 2-4 nodes were constructed using combinations of 45°, 90°, 135°, and 180° angles. Each system configuration was calibrated using both optimized and non optimized version of the calibration procedure. The validation object was presented to each configuration at ten locations within the scene. The reconstructed object was segmented from each scene by fitting a model to the calibrated frame. Once the object is successfully segmented its geometrical properties are measured and recorded. The resulting error of the reconstructed object was measured by comparison to the ground truth of the validation object.

5.2.1 Results from Calibration Experiments

The primary interest of the first calibration experiment is assessing the impact of widening the angle between a pair of cameras on the resulting precision of the reconstructed scene. The angle between the pair begins at 45° and is increased in 45° increments up to 180°. For each angle width the validation object was observed and reconstructed from ten locations. Since the validation object is spherical, all geometric properties are related to the radius. For this reason the radius is used as the recovered measurement. This sets the foundation for analyzing the results of this research because each calibrated frame is a conglomerate of camera pairs. The (absolute) mean error describes the expected number of millimeters the measured values differed from the ground truth. This is calculated using

$$\epsilon_{mean} = \frac{\sum_{i=1}^{i=n} |x_i - x_{groundtruth}|}{n}.$$

The percent error describes the amount of difference between the mean value measured and the ground truth. This is calculated using

$$x_{mean} = \frac{\sum_{i=1}^{i=n} x_i}{n}.$$

$$\epsilon_{percent} = \frac{x_{mean} - x_{groundtruth}}{x_{groundtruth}} * 100.$$

Table 5.1: Effect of Widening the Angle Between Pairs of Cameras

Angle	$\epsilon_{percent}$ (N.O)	ϵ_{mean}(N.O)	$\epsilon_{percent}$ (O)	ϵ_{mean} (O)
45°	4.31%	5.13 mm	2.92%	3.48 mm
90°	4.36%	5.19 mm	2.95%	3.51 mm
135°	5.68%	6.70 mm	3.12%	3.72 mm
180°	4.35%	5.18 mm	2.98%	3.55 mm

This table displays the percent error and mean error (ϵ_{mean}) calculated from the measurements of the validation object segmented from camera pairs calibrated with the optimized (O), and non-optimized (N.O) version of the calibration procedure.

The results presented in Table 5.1 do not fully describe the consistency of the recovered measurements. That is, the variance of the recovered error does not discriminate against measurements that are larger and smaller than the ground truth. In each region of the scene the recovered measurement should be comparable to that in any other viewable region. Hence, the need to evaluate the standard deviation of the measured values for each configuration. This is calculated using

$$deviation = \sqrt{\frac{\sum_{i=1}^{i=n}(x_i - x_{mean})^2}{n-1}}.$$

Table 5.2: Consistency of Measurements from Pairs of Cameras

Angle	Standard Deviation (N.O)	Standard Deviation (O)
45°	0.17 mm	0.15 mm
90°	0.12 mm	0.11 mm
135°	0.10 mm	0.10 mm
180°	0.11 mm	0.10 mm

This table displays the standard deviation of the measured validation object for camera pairs calibrated with the optimized (O), and non-optimized (N.O) version of the calibration procedure.

The next calibration experiment assesses the effect of combining pairs of calibrated cameras. This is achieved using three cameras, where one camera is common in both pairs. This camera's coordinate system was used as the global coordinate system and is assumed to be the primary camera. One pair of cameras is held fixed at 45°, while the other pair is increased from 45° to 135°. Next, the fixed pair is widened to 90°, and the other pair is started at 90° then widened to 180°. The same procedure for recovering and measuring the validation object was implemented in this experiment.

The percent error and mean error for optimized and non optimized calibration is presented in Table 5.4.

Table 5.3: Effect of Combining Two Pairs of Calibrated Cameras

Angle	$\epsilon_{percent}$ (N.O)	ϵ_{mean} (N.O)	$\epsilon_{percent}$ (O)	ϵ_{mean} (O)
45° - 45°	2.96%	3.527 mm	2.29%	2.372 mm
45° - 90°	3.14%	3.745 mm	2.34%	2.418 mm
45° - 135°	3.66%	4.637 mm	2.52%	2.546 mm
90° - 90°	3.53%	3.942 mm	2.39%	2.453 mm
90° - 180°	3.23%	3.582 mm	2.35%	2.420 mm

This table displays the percent error and mean error (ϵ_{mean}) calculated from the measurements of the validation object segmented from a frame consisting of calibrated pairs of cameras. The pairs that make up each frame were calibrated with the optimized (O), and non-optimized (N.O) version of the calibration procedure.

The concern raised by the variation in the recovered object's measurement was revisited with the addition of another camera. It is imperative to evaluate the consistency of the estimated measurements across pairs. Although the two child nodes of these systems are not calibrated with each other, the objects measured in the region observable by both child nodes should be comparable to any other viewable region. The standard deviation for optimized and non optimized calibration is presented in Table 5.4.

Table 5.4: Consistency of Measurements from Combined Pairs

Angle	Standard Deviation (N.O)	Standard Deviation (O)
45° - 45°	0.07 mm	0.05 mm
45° - 90°	0.04 mm	0.04 mm
45° - 135°	0.05 mm	0.03 mm
90° - 90°	0.04 mm	0.03 mm
90° - 180°	0.03 mm	0.04 mm

This table displays the standard deviation of the measured validation object for frames consisting of two combined calibrated pairs. Each pair was calibrated with the optimized (O), and non-optimized (N.O) version of the calibration procedure.

The final calibration experiment assesses the effect of combining pairs in which each camera has a child node. This is achieved using four cameras in three pairs. One pair contains the primary cameras from the other two pairs. One of the camera's coordinate system from this pair is arbitrarily selected as the global coordinate system. The angles between the cameras are varied to show the implications of combining different variations into a system frame. The same procedure for recovering and measuring the validation object was implemented in this experiment. The mean error and percent error for optimized and non optimized calibration is presented Table 5.5.

Table 5.5: Effect of Combining Cameras with Children

Angle	$\epsilon_{percent}$ (N.O)	ϵ_{mean} (N.O)	$\epsilon_{percent}$ (O)	ϵ_{mean} (O)
45° - 45° - 90°	1.09%	1.35 mm	1.06%	0.93mm
45° - 90° - 135°	1.23%	1.55 mm	1.08%	0.94 mm
45° - 45° - 135°	1.17%	1.41 mm	1.08%	0.93 mm
90° - 90° - 90°	1.16%	1.43 mm	1.07%	0.94 mm
90° - 90° - 135°	1.24%	1.77 mm	1.09%	0.96 mm
90° - 90° - 180°	1.16%	1.76 mm	1.08%	0.96 mm

This Table displays the percent error and mean error calculated from the measurements of the validation object segmented from a frame consisting of calibrated pairs of cameras that each have children. The pairs that make up each frame were calibrated with the optimized (O), and non-optimized (N.O) version of the calibration procedure.

Again, the concern raised by the variation in the recovered object's measurement was revisited. The particular regions of interest were viewable by cameras which were not calibrated together. These regions should yield results similar to those regions viewable by cameras that were calibrated together.

Table 5.6: Consistency of Measurements from Pairs with Children

Angle	Standard Deviation (N.O)	Standard Deviation (O)
45° - 45° - 90°	0.02 mm	0.01 mm
45° - 90° - 135°	0.03 mm	0.03 mm
45° - 45° - 135°	0.02 mm	0.02 mm
90° - 90° - 90°	0.01 mm	0.02 mm
90° - 90° - 135°	0.03 mm	0.01 mm
90° - 90° - 180°	0.01 mm	0.01 mm

This table displays the standard deviation of the measured validation object for frames consisting of two combined calibrated pairs that each have children. Each pair was calibrated with the optimized (O), and non-optimized (N.O) version of the calibration procedure.

5.3 Description of Sustaining Calibration Experiments

For the following experiments a calibrated pair of cameras observed the validation object for an hour. During observation there is no interference with the camera. On each frame the calculated updating transformation to each camera is computed, and the resulting data is aligned. The error in the updated data of each frame, and the frame aligned using the original transformation is then compared.

5.3.1 Results from Sustaining Calibration Experiments

The primary goal of the sustaining calibration experiment is assessing the algorithm's ability to sustain the precision achieved from the calibration phase. The scene and cameras are static such that accumulating error over long video sequences can be evaluated. First, attempts were made to successfully sustain alignment for an hour using simply feature matching. Features were required to be reciprocal and one to one. The maximum allowed distance between features was varied from 1 cm to 50 cm. Unfortunately, this method became uncalibrated relatively quickly. Table 5.7 presents an assessment of the amount of time to divergence of calibration versus the distance between corresponding features.

Table 5.7: Sustaining Calibration by Feature Matching

Max Distance Between Correspondences	Time Before Divergence
50 cm	42 seconds
25 cm	48 seconds
10 cm	66 seconds
5 cm	72 seconds
1 cm	90 seconds

This table displays the amount of time feature matching was able to sustain calibration for various distances between correspondences. The length of time for sustaining calibration to be considered successful was an hour.

Similarly, attempts were made to successfully sustain alignment for an hour using ICP. The maximum allowed distance between matching points was varied from 50 cm to 1 cm. This method also diverged eventually; however, the amount of time until divergence was substantially longer. Table 5.8 presents an assessment of the amount of time to divergence versus the distance between corresponding features.

The last part of this experiment was testing the method of combining feature matching and ICP described in section 4.3.2. This method was able to sustain calibration for an hour. For set time intervals the error from measuring the validation object is recorded using the update transformation and the original transformation. The results are graphed in Figure 5.1.

Table 5.8: Sustaining Calibration by ICP

Max Distance Between Correspondences	Time Before Divergence
50 cm	17.7 minutes
25 cm	22.8 minutes
10 cm	24.0 minutes
5 cm	27.8 minutes
1 cm	29.52 minutes

This table displays the amount of time ICP was able to sustain calibration for various distances between correspondences. The length of time for sustaining calibration to be considered successful was an hour.

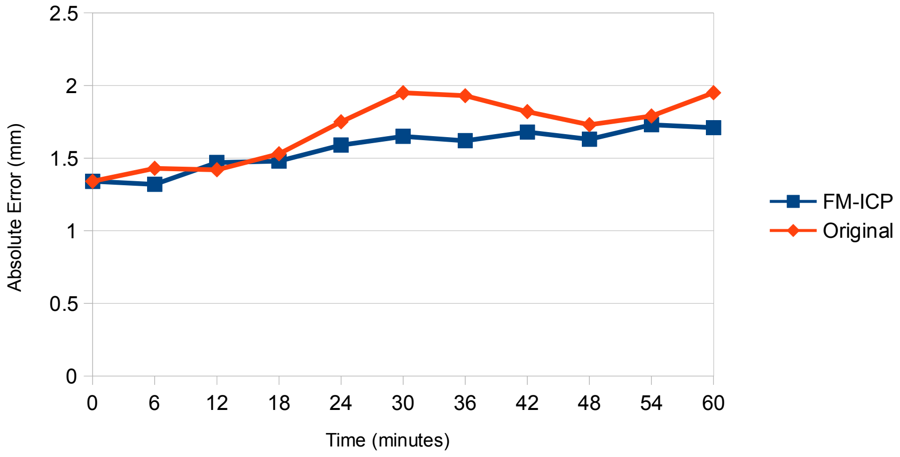


Figure 5.1: Sustaining Calibration by Feature Matching Combined with ICP. A pair of calibrated cameras observe a static scene for an hour while updating the transformation computed during calibration. Every 6 minutes the absolute error of reconstructing the validation object using the original transformation from the calibration, and the updated transformation from FM-ICP is graphed

5.4 Description of Maintaining Calibration Experiments

For the following experiments a calibrated pair of cameras observed a scene while random movements were induced on the camera (to simulate jitter). Once the updating transformation for a jitter is computed, the validation object is measured in various regions of the scene. The mean error of measurements recorded after a jitter is reported. This process is repeated ten times. The successive experiment jitters both cameras in a pair and the performs the assessment described for jittering one camera.

5.4.1 Results from Maintaining Calibration Experiments

The primary interest of the the first phase of this experiment is assessing the impact of jitter on the accuracy of the reconstructed scene. The mean error is computed using the same formula from the prior experiments. After a jitter, the error of the reconstructed validation object is measured in ten locations. The results are graphed in Figure 5.2.

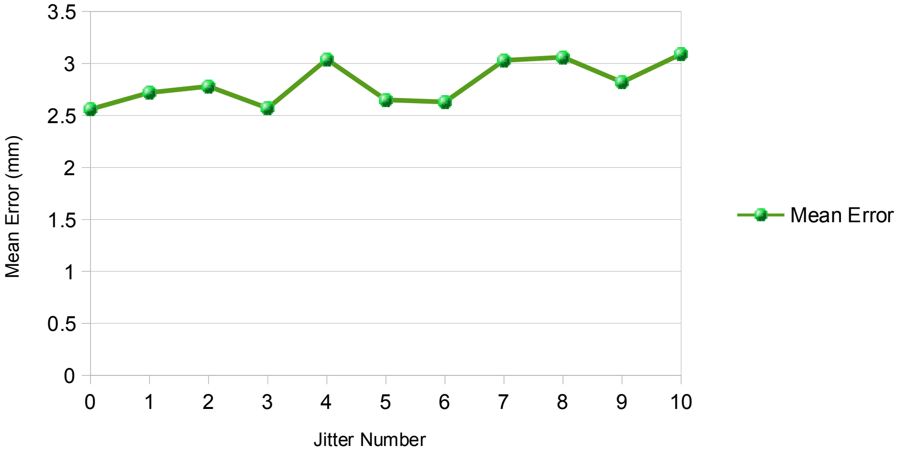


Figure 5.2: Effect of Jittering a Camera in a Calibrated Pair. During observation one camera in a calibrated pair is jittered ten times. After each jitter the absolute error in the reconstruction of the validation object is measured in ten locations. The (absolute) mean error of reconstruction after each jitter is plotted.

The primary interest of the second phase of this experiment is assessing the impact of jittering both cameras in a calibrated pair. Since the error in a frame is the sum of the error between frames, analyzing how jitter affects the accuracy of a pair gives insight to how the entire system of pairs will be affected. The mean error is computed using the same formula from the prior experiments. After a jitter to both cameras, the error of the reconstructed validation object is measured in ten locations.

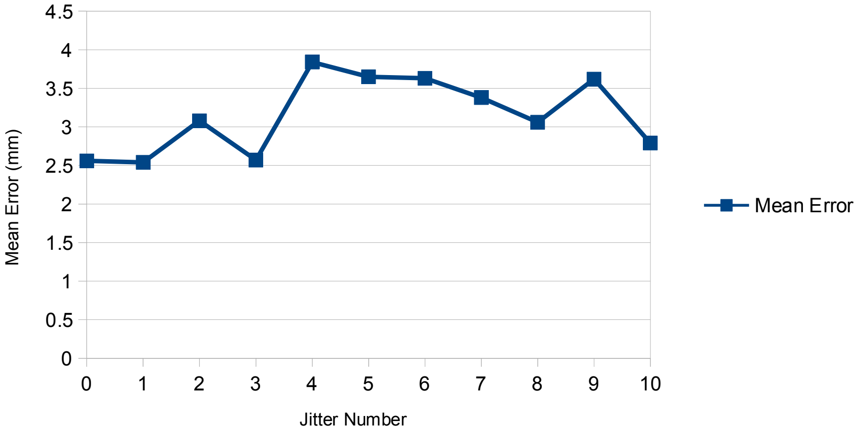


Figure 5.3: Effect of Jittering Both Cameras in a Calibrated Pair. During observation both cameras in a calibrated pair are jittered ten times. After each jitter the absolute error in the reconstruction of the validation object is measured in ten locations. The (absolute) mean error of reconstruction after each jitter is plotted.

6 DISCUSSION AND FUTURE WORK

6.1 Discussion

The results presented in the previous chapter support the claim of the presented procedure being effective, accurate, and plausible. The foundation of this procedure is the derivation of the alignment tree, which defines the underlying structure of the frame produced by the system of cameras. The alignment tree maximizes the potential accuracy of the reconstructed scene by finding the configuration in which the total overlap between all pairs is maximal. Using this structure also allows cameras to be isolated or grouped to achieve the goals of the application.

6.1.1 Conclusions on Achieving Calibration

The baseline of accuracy evaluation is between a single pair of cameras. Table 5.1 shows an increase of 24% increase in error when the non optimized calibration procedure is used and the angle between the cameras is increased from 45° to 135°. When the optimized calibration procedure is used there is only a 6% increase for the same interval. These results support the notion of pairing cameras with greater amounts of overlap. Moreover, there is an average of 30% reduction of error when the optimized calibration procedure is used. This supports the necessity of using the optimization criterium in order to achieve more accurate results. Conversely, the measurements obtained when the angle between the cameras was 135° had a deviation of approximately 70% less than that of 45°. This can be attributed to the fact that the larger angle yields a larger field of vision; hence, the object would be representative in more regions when viewed by cameras separated by the wider angle.

Table 5.3 shows a 30% decrease in error when adding a third camera and calibrating using the non optimized procedure. When the optimized procedure is used the error decreases by 46% when a third camera is added. The effect of joining pairs of cameras needed to be assessed before any strong claims about the relevance of maximizing the amount of overlap can be made. Table 5.3 shows an increase of 30% when combining a 45° pair with a 135° pair vs. a 45° pair combined with another 45° pair. When the non optimized version of the calibration procedure was used the increase was only 7%. There was about an 29% reduction using the optimization. When comparing the mean error reported for the optimized calibration procedures in table 5.1 and 5.3 there is an 31% reduction in the error when adding a third camera. It is believed that the added portion of the scene added mediates discrepancies realized by just two cameras.

Table 5.5 solidifies that combining pairs of cameras with the maximal overlap yields the more accurate results. However, there is only an 7% reduction in error when using the optimized calibration procedure. This suggests that the benefit of optimization is overshadowed by model completeness. More importantly, table 5.6 shows that deviation of measurements are relatively similar for configurations that can view the same amount of a scene. This supports the claim that a more full 3D model will yield consistent measurements in all visible regions of the scene. This supports the claims of this procedure being consistent.

6.1.2 Conclusions on Maintaining Calibration

Initially, the major concern with maintaining calibration was accounting for possible misalignment. During construction of the outlined procedure the issue of accumulated error from approximating movement of cameras became a prominent concern as this side effect is magnified by the use of multiple cameras. This lead to the evaluation of the effect of error propagation over time of usage of the procedure. Tables 5.7 and 5.8 show that feature matching and ICP implemented independently are not effective at sustaining calibration overtime. Using the revised ICP algorithm outlined in section, calibration was able to be sustained for an hour. Figure 5.1 shows that over the course of an hour the increase in absolute error was at most 0.5 mm more than the initial error. When using the original calibration the error recovered was almost double that. This is due purely to the mechanics of the camera and the inherent noise introduced form the point cloud formalization.

The major assumptions behind correcting calibration is that jitter may happen occasionally, and movement is much slower than the speed images are captured. The goal is to maintain relative accuracy through unintended shifts in camera position. The major challenge was finding the best threshold to discriminate noisy stationary data from potential small movement between frames. Figure 5.2 shows that the mean error recorded after induced jitters was within 0.5 mm of the initial error. This suggests that the effects of occasionally jittering a single camera is minimal. Figure 5.3 shows that jittering both cameras in a pair yields a slightly larger margin of error; however, this is acceptable since jittering is assumed to happen rarely.

6.2 Future Work

The overarching goals for future work are to achieve real time functionality, increase the scope of applicable systems of overlapping cameras, and achieve compatibility with dynamic environments. Achieving these goals will empower the procedure discussed in this thesis to be functional in a real world application. A discussion on each goal is provided below.

6.2.1 Real Time Functionality

Increasingly in research GPUs are being used to process highly parallelizable routines in the interest of speeding up computation. Works such as [7] show that offloading the iterative processing of alignment algorithms to GPUs allows for real time processing of point clouds. In this thesis all processing was performed off-line and required several minutes to complete computation of a single calibrated frame. Also, each computer had multiple camera connections; the data generated had to be processed from each connected camera sequentially. These factors are undoubtably the biggest challenge for achieving real time processing. However, the processing of the alignment algorithms are parellelizable, as well as the processing of the data generated from cameras sharing a computer. This introduces multiple places in which employing GPU processing may be beneficial, but the biggest concern is in the number of GPUs needed to process data from multiple cameras.

6.2.2 Increasing Scope

The interest in increasing the scope of this procedure still pertains to low overlapping groups of cameras. The environments worked with in this thesis focused on camera pointing to the same point. Contrastingly, pointing the cameras outward while still maintaing low overlap between views is outside the scope of this thesis. Intuitively, the procedure described in this thesis should be compatible with such a group of overlapping cameras because the calibrating procedure uses an inserted set of features, which are only present in the region of overlap. The direction the cameras are facing are irrelevant to the outlined procedure.

6.2.3 Compatibility with Dynamic Environments

Environments used in this thesis were static; however, cameras were allowed to move. The movement of the cameras was used to update the location of the data in the reconstructed frame. If an object were to move spurious results may be recovered. The procedure would then have to be grown to account for discrepancies introduced by moving objects. Ideally, the system should be able to account for multiple moving objects, and a jittering camera. A sense of background objects (stationary) and foreground objects (dynamic) may be useful in keeping track of independent movements.

REFERENCES

[1] A.W. Fitzgibbon. ”Robust Registration of 2D and 3D Point Sets.” *Proceedings from British Machine Vision Conference*, pp. 662-670, 2001.

[2] Chetverikov, Dmitry, et al. ”The trimmed iterative closest point algorithm.” *Pattern Recognition, 2002. Proceedings. 16th International Conference on.* Vol. 3. IEEE, 2002.

[3] Caner, G.; Tekalp, A.M.; Sharma, G.; Heinzelman, W. ”Multi-View Image Registration for Wide-Baseline Visual Sensor Networks.” *Image Processing, 2006 IEEE International Conference on*, pp.369-372, 8-11 Oct. 2006.

[4] Chetverikov, Dmitry, Dmitry Stepanov, and Pavel Krsek. ”Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm.” *mage and Vision Computing 23.3* (2005): 299-309.

[5] Ce Li; Jianru Xue; Shaoyi Du; Nanning Zheng. ”A Fast Multi-Resolution Iterative Closest Point Algorithm.” *Pattern Recognition (CCPR)*, 2010 Chinese Conference on Pattern Recognition, pp.1-5, 21-23 Oct. 2010.

[6] Hao Du , Peter Henry , Xiaofeng Ren , Marvin Cheng , Dan B. Goldman , Steven M. Seitz , Dieter Fox. ”Interactive 3D modeling of indoor environments with a consumer depth camera.” *Proceedings of the 13th international conference on Ubiquitous computing*, September 17-21, 2011, Beijing, China

[7] Newcombe, Richard A., et al. ”KinectFusion: Real-time dense surface mapping and tracking.” *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on.* IEEE, 2011.

[8] Jihua Zhu; Nanning Zheng; Zejian yuan; Shaoyi Du. ”Point-to-line metric based Iterative Closest Point with bounded scale.” *Industrial Electronics and Applications.* 2009. ICIEA 2009. 4th IEEE Conference on , pp.3032-3037, 25-27 May 2009.

[9] Langis, C.; Greenspan, M.; Godin, G. ”The parallel iterative closest point algorithm.” *3-D Digital Imaging and Modeling.* 2001. Proceedings. Third International Conference on , pp.195-202, 2001.

[10] Lingyun Liu; Gene Yu; Wolberg, G.; Zokai, S. ”Multiview Geometry for Texture Mapping 2D Images Onto 3D Range Data.” *Computer Vision and Pattern Recognition.* 2006 IEEE Computer Society Conference, pp. 2293- 2300, 2006.

[11] Puwein, J.; Ziegler, R.; Vogel, J.; Pollefeys, M. ”Robust multi-view camera calibration for wide-baseline camera networks.” *Applications of Computer Vision (WACV).*2011 IEEE Workshop on, pp.321-328, 5-7 Jan. 2011.

[12] Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. ”Aligning point cloud views using persistent feature histograms.” *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on.* , pp.3384-3391, 22-26 Sept. 2008.

[13] Rusu, Radu Bogdan; Blodow, Nico; Beetz, Michael. ”Fast Point Feature Histograms (FPFH) for 3D registration.” *Robotics and Automation.* 2009. ICRA ’09. IEEE International Conference on, pp.3212-3217, 12-17 May 2009.

[14] Yang, Michael, Yanpeng Cao, Wolfgang Forstner, and John McDonald. ”Robust wide baseline scene alignment based on 3d viewpoint normalization. ”*Advances in Visual Computing.* (2010): 654-665.

[15] Paton, Michael, and J. Kosecka. ”Adaptive RGB-D Localization.” *Computer and Robot Vision (CRV).* 2012 Ninth Conference on, pp. 24-31. IEEE, 2012.

[16] Sakai, S.; Ito, K.; Aoki, T.; Unten, H. ”Accurate and dense wide-baseline stereo matching using SW-POC.” *Pattern Recognition (ACPR)*, 2011 First Asian Conference on, pp.335-339, 28-28 Nov. 2011.

[17] Martinec, Daniel, and Tom Pajdla. ”Consistent multi-view reconstruction from epipolar geometries with outliers.” *Image Analysis*, (2003): 477-486.

[18] Gumhold, Stefan, Xinlong Wang, and Rob MacLeod. ”Feature extraction from point clouds.” *In Proceedings of 10th international meshing roundtable, vol. 2001.* 2001.

[19] Seitz, S.M.; Curless, B.; Diebel, J.; Scharstein, D.; Szeliski, R. ”A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms.” *Computer Vision and Pattern Recognition.* 2006, pp. 519- 528, 17-22 June 2006.

[20] Strecha, C.; von Hansen, W.; Van Gool, L.; Fua, P.; Thoennessen, U. ”On benchmarking camera calibration and multi-view stereo for high resolution imagery.” *Computer Vision and Pattern Recognition.* 2008. pp.1-8, 23-28 June 2008.

[21] Pons, Jean-Philippe, Renaud Keriven, and Olivier Faugeras. ”Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score.” *International Journal of Computer Vision 72, no. 2.*, (2007): 179-193.

[22] Dang, Thao, Christian Hoffmann, and Christoph Stiller. ”Continuous stereo self-calibration by camera parameter tracking.” *Image Processing, IEEE Transactions on 18.7.* (2009): 1536-1550.

[23] Graham, Ronald L., and Pavol Hell. "On the history of the minimum spanning tree problem." *Annals of the History of Computing 7.1.* (1985): 43-57.

[24] Yanpeng Cao; Yang, M.Y.; McDonald, J. "Robust alignment of wide baseline terrestrial laser scans via 3D viewpoint normalization." *Applications of Computer Vision (WACV).* 2011 IEEE Workshop on, pp.455-462, 5-7 Jan. 2011.

[25] P. Henry et al. "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments." *In Proc. of the Int. Symposium on Experimental Robotics (ISER)*, 2010.

[26] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera." *In Proceedings of the IEEE CVPR*, 2010.