Real-Time Object Motion and 3D Localization from Geometry

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy
in the Graduate School of The Ohio State University

By
Young Jin Lee

Graduate Program in Geodetic Science and Surveying

The Ohio State University
2014

Dissertation Committee:
Alper Yilmaz, Advisor
Alan Saalfeld
Ralph R. B. von Frese
Abstract

Knowing the position of an object in real-time has tremendous meaning. The most widely used and well-known positioning system is GPS (Global Positioning System), which is now used widely as invisible infrastructure. However, GPS is only available for outdoor uses. GPS signals are not available for most indoor scenarios. Although much research has focused on vision-based indoor positioning, it is still a challenging problem because of limitations in both the vision sensor itself and processing power. This dissertation focuses on real-time 3D positioning of a moving object using multiple static cameras. A real-time, multiple static camera system for object detection, tracking, and 3D positioning that is run on a single laptop computer was designed and implemented. The system successfully shows less than ±5 mm in real-time 3D positioning accuracy at an update rate of 6 Hz to 10 Hz in a room measuring 8×5×2.5 meters. Implementation and experimental analysis has demonstrated that this system can be used for real-time indoor object positioning.

In addition, ‘collinearity condition equations of motion’ were derived that represent the geometric relationship between 2D motions and 3D motion. From these equations, a ‘tracking from geometry’ method was developed that combines these collinearity condition equations of motion with an existing tracking method to simultaneously estimate 3D motion as well as 2D motions directly from the stereo camera system. A
A stereo camera system was built to test the proposed methods. Experiments with real-time image sequences showed that the proposed method provides accurate 3D motion results. The calculated 3D positions were compared with the results from an existing 2D tracking method that uses space intersection. The differences between results of the two methods were less than ±0.01 mm in all X, Y, and Z directions. The advantage of the tracking from geometry method is that this method calculates 2D motions and 3D motion simultaneously, while other tracking methods need an additional triangulation step to estimate 3D positions.
Dedication

To my family
Acknowledgments

I would like to thank Dr. Alper Yılmaz, my advisor, for all his help, guidance, and patience for not only academic life but also my personal life. Also, I would like to thank Dr. Woosug Cho, who was my advisor in my master’s study, for his encouragement and guidance to this field as well as to OSU. I would like to express my gratitude to the members of my dissertation committee, Dr. Alan Saalfeld and Dr. Ralph R. B. von Frese. I also would like to thank my fellows in the OSU Photogrammetric Computer Vision Laboratory and other colleagues in Geodetic Science and Surveying. Finally, with all my heart, thank you to my family.
Vita

2001 ............................................................ B.E. Civil Engineering, Inha University, Incheon, Korea,

2003 ............................................................ M.E. Geographic Information Engineering, Inha University, Incheon, Korea,

2010 ............................................................ M.S. Geodetic Science & Surveying, The Ohio State University,

2007 to present ........................................... Graduate Research Associate, Geodetic Science & Surveying, The Ohio State University

Publications


Fields of Study

Major Field: Geodetic Science and Surveying
Table of Contents

Abstract................................................................................................................................................. ii

Dedication ........................................................................................................................................... iv

Acknowledgments .............................................................................................................................. v

Vita .................................................................................................................................................... vi

Publications .......................................................................................................................................... vi

Table of Contents ............................................................................................................................... viii

List of Tables ........................................................................................................................................ xii

List of Figures ...................................................................................................................................... xiii

1. INTRODUCTION .......................................................................................................................... 1

1.1 Problem Definition ...................................................................................................................... 1

1.1.1 Real-Time Multiple Camera Object Detection, Tracking, and 3D Positioning System .......... 1

1.1.2 Motion from Geometry ............................................................................................................ 3

1.2 Goals .......................................................................................................................................... 6

1.3 Related Research ......................................................................................................................... 7

1.4 Organization ................................................................................................................................ 13
2. BACKGROUND ........................................................................................................... 14

2.1 The Gauss-Markov Model .................................................................................. 14

2.1.1 The LESS (Least-Squares Solution) for the GMM ....................................... 15

2.2 Photogrammetry ............................................................................................... 17

2.2.1 Collinearity Condition Equations .................................................................. 17

2.2.2 Interior Orientation ....................................................................................... 21

2.2.3 Exterior Orientation ...................................................................................... 22

2.2.4 Camera Calibration ....................................................................................... 23

2.2.5 Coordinate Transformations .............................................................. 26

2.2.6 Space Intersection .......................................................... 29

2.2.7 Space Intersection with Multiple Cameras ................................................. 32

2.2.8 Space Resection ....................................................................................... 33

2.2.9 Bundle Adjustment .................................................................................... 35

2.2.10 Epipolar Geometry ................................................................................. 37

2.3 Object Tracking ............................................................................................... 38

2.3.1 Optical Flow Constraint Equation .............................................................. 38

2.3.2 Lucas-Kanade Feature Tracker ..................................................................... 41

2.3.3 Image Matching ......................................................................................... 44
3. REAL-TIME MULTIPLE CAMERA OBJECT DETECTION, TRACKING, AND 3D POSITIONING SYSTEM

3.1 System Configuration

3.1.1 Interior Orientation of Cameras

3.1.2 Target and Camera Installation

3.1.3 Exterior Orientation

3.1.4 Time Synchronization of Cameras and Frame Rate

3.2 Detection, Tracking and Refinement

3.2.1 The Target Object

3.2.2 Detection

3.2.3 Tracking

3.2.4 Refinement

3.3 Experiment

3.3.1 Detection, Tracking, and Refinement

3.3.2 Real-Time Bit Tip Positioning

3.4 Conclusion

4. MOTION FROM GEOMETRY

4.1 Collinearity Condition of Motion

4.1.1 Collinearity Condition Equations of Motion
List of Tables

Table 2.1. Interior orientation parameters used for Figure 2.3 ........................................ 25
Table 2.2. Space intersection algorithm ......................................................................... 31
Table 2.3. Lucas-Kanade feature tracker algorithm ......................................................... 43
Table 3.1. Estimated interior orientation parameters for cameras used in the system ......51
Table 3.2. Estimated exterior orientation parameters for CAM0, CAM1, CAM2, and CAM3 ........................................................................................................................... 54
Table 3.3. Classification of a match [73] ........................................................................ 61
Table 4.1. The estimated interior orientation parameters for CAM4 and CAM5 ............. 91
Table 4.2. The estimated exterior orientation parameters for CAM4 and CAM5 .......... 92
Table 4.3. Simulation data for space intersection of motion ........................................... 93
Table 4.4. Comparison of 3D positions estimated by 2D tracking (Lucas-Kanade then triangulation) method and 3D tracking (tracking from geometry) method at the last frame ........................................................................................................ 96
List of Figures

Figure 1.1. Conceptual diagram for a real-time multiple static camera object detection, tracking, and 3D positioning system ................................................................. 2

Figure 1.2. Motion in image plane .................................................................................. 3

Figure 1.3. Conceptual diagram for estimating 3D position of a 3D object (A) from 2D position observations (a1 and a2) in multiple images (L1 and L2) ................................. 4

Figure 1.4. Conceptual diagram for ‘tracking then triangulation’ (left) and ‘tracking from geometry’ (right), which can directly provide 3D motion ........................................... 5

Figure 2.1. Collinearity condition; a point in object space (A), its image on a photograph (a), and the perspective center of the photograph (L) are on a straight line in three-dimensional space .................................................................................. 18

Figure 2.2. Exterior orientation parameters (EOP) represent position and orientation of the sensor (camera) in 3D object space. ................................................................. 22

Figure 2.3. Lens distortion patterns: symmetric radial (left), decentering (right), and combined symmetric radial and decentering (bottom); distortion magnitudes are exaggerated for visualization. Table 2.1 shows interior orientation parameters for these patterns ................................................................................................................. 24

Figure 2.4. Coordinate transformations between the photo coordinate system and the image coordinate system .......................................................................................... 26

Figure 2.5. Conceptual diagram of space intersection: the space intersection method estimates 3D coordinates of a point in the object space from photo coordinates observations in a stereo pair ................................................................. 29

Figure 2.6. Space intersection with multiple cameras ..................................................... 33

Figure 2.7. Conceptual diagram of space resection: the space resection method estimates the exterior orientation parameters of a photograph using known control points .......... 33
Figure 2.8. Conceptual diagram of bundle adjustment: the bundle adjustment estimates all photogrammetric observations simultaneously using control points. .........................35

Figure 2.9. Conceptual diagram of epipolar geometry....................................................37

Figure 2.10. The optical flow constraint equation...........................................................40

Figure 2.11. Template matching finds parts of original image (left) that correspond to a template (middle). The matching result is marked by a red rectangle in the right.........44

Figure 2.12. Conceptual diagram of template matching..................................................46

Figure 2.13. Conceptual diagrams of epipolar constraint (left) and vertical line locus constraint (right) for image matching: epipolar constraint reduces search area in a single epipolar line while vertical line locus constraint reduces the search area on the epipolar line. ...............................................................................................46

Figure 2.14. Reducing search space with the epipolar constraint in template matching...47

Figure 2.15. Reducing search space with vertical line locus constraint on an epipolar line .......................................................................................................................48

Figure 3.1. System configuration of multiple static camera real-time object detection, tracking, and 3D positioning system [73] .................................................................50

Figure 3.2. AVT Manta G-125 Camera with Kowa LM5NCL Lens ...............................51

Figure 3.3. Distribution of camera positions (left) and detected control points (right top) for camera calibration with lens distortion patterns of CAM0 (right bottom) ..........52

Figure 3.4. Distribution of 82 control points in an 8×5×2.5 m room that were used to determine the exterior orientation parameters of the cameras in the system. .................53

Figure 3.5. Installed cameras: (from left to right) CAM0, CAM1, CAM2, and CAM3 ...53

Figure 3.6. Camera exterior orientation results (focal length and size of image plane are exaggerated 50 times for visualization) ..........................................................................55

Figure 3.7. Target object (yellow box) and its templates [73]........................................57
Figure 3.8. Epipolar correspondence used to find the correct match among candidates ..58

Figure 3.9. Detection, tracking and refinement workflow of a camera .........................60

Figure 3.10. Legend for the screenshots: camera ID (left) and information on a single camera screen (right) [73] ........................................................................................................63

Figure 3.11. Case 1 [73]: two cameras are in **tracking** mode and two cameras are in **detection** mode; two wrong matches are automatically corrected by refinement process.64

Figure 3.12. Case 2 [73]: all cameras are in **tracking** mode; however, only two correct observations are automatically selected to be used for bundle adjustment. ...............65

Figure 3.13. Case 3 [73]: all cameras are in **tracking** mode; however, there is only one correct match, therefore, bundle adjustment is not available. ........................................66

Figure 3.14. Case 4 [73]: all cameras are in **tracking** mode; refinement process finds two correct matches and correct two wrong matches using bundle adjustment result. ........67

Figure 3.15. Trajectory of BWB within 1022 frames; color represents time increment...68

Figure 3.16. Mock-up for accuracy and precision test ..................................................69

Figure 3.17. Real-time BWB 3D position estimation at Location 2 .........................70

Figure 3.18. Real-time BWB positioning accuracy (difference) ..............................71

Figure 3.19. Real-time BWB position precision (standard deviation) .......................71

Figure 3.20. Conceptual diagram of drill bit tip 3D positioning ................................72

Figure 3.21. Cone-shaped target installed for real-time accuracy evaluation ..........73

Figure 3.22. An example of real-time drill bit tip positioning accuracy test .............74

Figure 3.23. Real-time drill bit tip position accuracy .................................................75

Figure 4.1. Conceptual diagram of the collinearity condition of motion .................78

Figure 4.2. Stereo camera system built for experiments ...........................................90
Figure 4.3. Image with lens distortion (left) and distortion corrected image (right) using the estimated lens distortion parameters.................................................................91

Figure 4.4. The position of the stereo camera system (blue) and trajectory of a moving object (red).........................................................................................................................93

Figure 4.5. Calculated 3D motion error with respect to frame number.................................94

Figure 4.6. Calculated 3D position error with respect to frame number .................................95

Figure 4.7. Target object for tracking from geometry method ..................................................96

Figure 4.8. Differences in calculated 3D position results between 2D tracking (Lucas-Kanade then triangulation) and 3D tracking (tracking from geometry) methods with respect to frame number ..................................................................................................................97

Figure 4.9. Tracking results at frame 189: 2D tracking (top) and 3D tracking (bottom)..98
Chapter 1 is organized as follows. The discussion starts with the problem definition for this dissertation. This discussion leads to the main goals of the dissertation. Then, related research is reviewed. Finally, organization of this dissertation is outlined.

1.1 Problem Definition

1.1.1 Real-Time Multiple Camera Object Detection, Tracking, and 3D Positioning System

Knowing the position of an object in real-time has tremendous meaning. The most widely used and well-known positioning system is GPS (Global Positioning System), which is now widely used as invisible infrastructure [1]. A GPS receiver estimates its position by using trilateration of distances between multiple GPS satellites and the receiver itself [2]. However, for most indoor scenarios, GPS signals are not available [3]. There has been much research focused on indoor positioning. Sensors used for indoor positioning include vision, ultrasound, radio signal, and inertial sensors [3]. Vision-based indoor positioning research can be categorized by the object to be located [4]. One category can be
positioning mobile vision sensors itself. Another can be positioning objects using static vision sensors. However, vision-based indoor positioning is still a challenging topic because of the limitations in both the vision sensor and processing power. In this dissertation, the focus is on real-time 3D positioning of a moving object using a multiple static camera system.

Figure 1.1. Conceptual diagram for a real-time multiple static camera object detection, tracking, and 3D positioning system

Using static cameras for real-time object positioning has a number of problems. First, the system should provide reliable 3D positioning accuracy in real-time. To achieve reliable 3D positioning accuracy, the size of the image is very important. However, using a large image can put burdens on processing resources. Furthermore, for real-time 3D positioning, images from multiple cameras need to be processed in real-time. Second, an object to be located should be view as well as scale invariant to all cameras, i.e., the
camera should recognize the object in any direction as well as at any distance. These problems were investigated and a real-time, multiple static camera system for object detection, tracking, and 3D positioning (Figure 1.1) was developed able to address these problems. Chapter 3 discusses this system and experiment results in detail.

1.1.2 Motion from Geometry

Tracking is defined as a method of estimating object trajectory in the image plane [5]. Assuming that there is a moving object and that the object is shown in consecutive images acquired from a camera at times $t_0$ and $t_1$, then tracking provides a motion in the image plane $(u, v)$ that represents displacement in the $x$ and $y$ directions (Figure 1.2). However, motion of interest happens in a real, three-dimensional, world. In other words, motion in 2D image space is just a projection of motion in the 3D world [6]. Therefore the goal of this research is to investigate calculating 3D motion directly from a set of 2D motions.
Assuming there is a moving object and its motion in 3D is to be calculated using multiple cameras, then 3D motion can be calculated indirectly using one of the existing 2D tracking methods followed by a 3D positioning method. Figure 1.3 illustrates the concept of estimating the 3D position of an object using a stereo camera setup. If the geometry of the stereo camera ($L_1$ and $L_2$) is known and an object ($A$) is shown in both cameras ($a_1$ and $a_2$), then the 3D position of the object can be estimated by the intersection two straight lines ($L_1a_1$ and $L_2a_2$).

![Figure 1.3. Conceptual diagram for estimating 3D position of a 3D object ($A$) from 2D position observations ($a_1$ and $a_2$) in multiple images ($L_1$ and $L_2$)](image)

Each line passes the camera center ($L_1$ or $L_2$) and the object position in each image ($a_1$ or $a_2$). This process is called space intersection and, as it is nonlinear, it therefore needs an iterative process to find the solution. Section 2.2.6 explains the photogrammetric space intersection method in detail.
Assuming one wants to estimate the 3D position of a moving object in image sequences acquired from a stereo camera, one can use a tracking method to find the motions of the moving object in 2D from each image sequence (Figure 1.4, left top) and then estimate the 3D position of the object at each time frame using the photogrammetric space intersection method (Figure 1.4, left bottom). However, if the geometry between the cameras is known, 3D motion \((AA')\) as well as 2D motions \((a_1a_1'\) and \(a_2a_2')\) can be directly calculated from image sequences using the ‘tracking from geometry’ method (Figure 1.4, right) developed in this dissertation research. To calculate 3D motion directly, first ‘collinearity condition equations of motion’ were developed that define the relationship between motion in 2D and motion in 3D. Then this was substituted into a tracking method to find 3D motion directly from image sequences. Chapter 4 discusses these concepts and experiment results in detail.
1.2 Goals

This dissertation research had two primary goals:

I. Implementing a real-time object detection, tracking, and 3D positioning system using multiple static cameras; and

II. Investigating the relationship between 2D motions and 3D motion, or ‘collinearity condition of motion’, which includes
   i. Demonstrating direct calculation of 3D motion as well as 2D motions from image sequences using the ‘tracking from geometry’ method.

The first goal of this dissertation research was implementing and demonstrating a real-time, multiple static camera system for object detection, tracking, and 3D positioning system able to run on a single laptop computer. Tracking and 3D positioning with multiple cameras in real-time is a challenging task considering the limitations of tracking and detection as well as the large amount of data from each camera. Investigation was made into minimizing these limitations to provide reliable 3D positioning solutions by fusing algorithms and methods from both the photogrammetry and computer vision fields.

The second goal was investigating the relationship between 2D motions in image space and 3D motion in object (or real world) space. A point in 3D and its image in a
photograph have a geometric relationship of the collinearity condition. In this dissertation, the ‘collinearity condition of motion’ will be shown. The collinearity condition equations of motion represent the geometric relationship between 3D motion of a point in 3D object space and 2D motion of its image in 2D photo space. A novel method to calculate the 3D motion directly in a multiple-camera system using the developed ‘tracking from geometry’ method that processes without triangulation was developed in this research.

1.3 Related Research

Real-time, multi-view object tracking is a challenging task because of its complexity and requisite processing speed. There is a large amount of research that has focused on either real-time object tracking or multiple-view object tracking. However, only a relatively small number of research studies have focused on the two topics together. Solving object detection, tracking, and 3D positioning simultaneously in a multiple, high-definition camera setup in real-time requires efficient algorithm development. Most 3D tracking methods find features in multiple views and then use triangulation to estimate its 3D position [7][8][9]. Despite its complexity, real-time, multiple-camera object tracking as well as 3D positioning has tremendous potential for applications in fields such as visual surveillance, object/pedestrian/animal monitoring, visual odometry, robot vision, SLAM (Simultaneous Localization and Mapping), and indoor positioning. For this dissertation
research, literature was reviewed that focused on real-time, multi view-based tracking and 3D positioning.

One of most active areas using multiple cameras for real-time tracking and 3D positioning is visual surveillance, or people tracking. In this field, solving problems due to occlusion between people is a very important issue [10]. Many researchers have used multiple cameras to solve the occlusion problem [11][12][13]. For instance, Krumm et al. [14] introduced a multi-person tracking system using two sets of stereo cameras. They performed tracking on three computers; two of which were used to process images from the stereo cameras and the third to combine the two independent results. The processing speed of their implementation was 3.5 frames per second (fps). Qu et al. [15] performed multiple-camera, multi-target tracking using a Bayesian framework to overcome the occlusion problems of multiple targets. Heath et al. [16] introduced a multi-person tracking system using multiple stereo cameras. They used a Harris detector to find feature points and then used the LKT method for tracking. Points were triangulated to estimate 3D positions. Tsutsuo et al. [17] tracked a person using multiple cameras. They assumed a person occupies a cylinder. Optical flow was used to find the moving person and then find the position of the assumed 3D cylinder. Mittal et al. [18][19] developed a stereo-camera people-tracking system employing a region-based stereo algorithm that does not need point matching to find the 3D position of a point. They segmented images from each camera and then generated an object-location likelihood map using 3D points from the
region-based stereo algorithm. They iteratively updated the position of objects in the ground plane using this map with occlusion analysis. Orwell et al. [20] introduced a multiple-camera tracking framework for surveillance that uses planar correspondence between 2D and 3D. They used homography transformation between an image and the ground plane to register objects. Tyagi et al. [7] proposed a tracking method based on 3D kernels that can track an object in 3D using multiple cameras. They initially selected a 3D volume of interest and then projected the volume to each camera to find the corresponding image region to be tracked.

Another active area that uses multiple views for tracking and 3D positioning is visual odometry [21][22]. There are a large number of research studies for robot navigation that have used vision sensors to estimate 3D feature location based on pose of the sensor platform [23][24]. Scaramuzza and Fraundorfer [25][26] summarized current visual odometry methods and their detection, tracking, and triangulation algorithms. Ni and Dellaert [27] presented a stereo camera tracking method for visual odometry. They detected points in one image with a Harris detector [28] and then found a match in the other image of the stereo pair using epipolar geometry. Then they used a modified Lucas-Kanade tracker to find the point in the next frame. Johnson et al. [29] introduced a visual odometry method that uses stereo images obtained by the Mars Exploration Rover (MER). They used a Harris detector to find interest points in the left image of a stereo pair and then found corresponding points using pseudo-normalized cross-correlation.
(PNC) template matching and epipolar constraint. They estimated 3D coordinates of the point using triangulation. They also used the same PNC method to find the interest points in the consecutive frame. Agrawal et al. [30][31] presented an autonomous localization system using stereo-vision odometry. They found Harris corner points in the stereo image, and then matched point pairs using normalized cross correlation (NCC) in both images with epipolar constraint. They limited the maximum speed of the platform so as to limit the search space to find the corresponding pairs in the next frame. Then they triangulated points with RANSAC [32] to find the 3D coordinates. Lin and Wang [33] presented a stereo-based SLAM system. They found feature points using the Shi-Tomasi method [34] and then applied KLT to track the feature points in one camera. The corresponding feature points were directly selected using stereo correspondence in the other camera. Oskiper et al. [35] designed a visual odometry system that uses a pair of stereo cameras (forward- and backward-looking). They found the Harris corner points and then triangulated the points to estimate their 3D positions. They used image matching to find 2D point correspondences between features in the previous and then the next frames. They estimated the pose of the left camera in each stereo system by space resection with RANSAC.

Researchers also have used multiple-camera object tracking for tracking and/or estimating the pose of objects [36][37][38]. Chu et al. [39] introduced a real-time, 3D, body-pose tracking method using images from a stereo IR (infrared) camera. In this
research, they extracted blobs in images and then matched the blobs to the 2D body model that is a projection of an iteratively updated 3D body model. In 2004, Caillette et al. [40][41] introduced a real-time, human-body tracking system. First, they generated voxels from multiple images using the statistical 3D reconstruction method. Then they used a model-based tracking method that finds correspondence between the blob position of the voxels and a kinematic model of the human body to estimate the motion of the body. Cai et al. [42] introduced a multiple-camera pose-estimation system. They used KLT to track interest points in each camera as detected by a Harris corner detector. Then they back-projected the interest points onto the mesh model to find their 3D location on the model. They estimated a change of pose by using the sum of projection errors. Choi et al. [43] estimated the pose of a rigid-body object using stereo cameras. They tracked SIFT [44] points using KLT with RANSAC and then calculate 3D points using stereo matching.

Car navigation or safety is one of the most active areas using multi-view object tracking [45][46]. For instance, Morat et al. [47] presented a stereo-vision tracking system for car navigation. They used the Lucas-Kanade method with epipolar and magnification constraints. The epipolar constraint prevents $y$-parallax while the magnification constraint maintains the ratio of the $x$-parallax of a tracked feature with respect to its previous position in 3D. They formulated two constraints into the Lucas-Kanade feature tracker, which is very similar to our ‘tracking from geometry’ method. Badino et al. [8]
introduced a moving-object detection method using a stereo camera that can be used for autonomous navigation. They used optical flows from each camera and combined them by using stereo geometry with a Kalman filter [48] to iteratively refine the velocity and position of a 3D point. Klappstein et al. [49] presented a vehicle-mounted, three-camera, moving-object detection system. They used KLT to track feature points and then triangulation to estimate the 3D coordinates. They used the estimated 3D motions from the triangulated feature points to segment moving objects. Lenz et al. [50] developed a moving-object detection method for car safety. They found interest points in two consecutive stereo image pairs and then calculated the 3D position. They connected the calculated 3D points using the Delaunay triangulation method to find a moving object by thresholding with calculated scene flow.

There also have been many research studies in the biomedical field focusing on real-time motion on flying insects, tracking them with multiple cameras [51][52][53]. Their interests are mainly in insects and their flying patterns, and their target space is, therefore, relatively smaller than those of other fields as well as being well controlled in illumination and background. For example, Grover and Tavare [54] introduced a real-time, insect 3D tracking system using three cameras. They constructed the 3D visual hull of flying insects using silhouettes from each camera and then tracked the insect using an extended Kalman filter. The image size of each camera was 640×480 pixels, and the tracking speed was 60 fps. However, their object space (a 25×75 mm vial containing live
drosophilae) was very small and controlled (blue background). Straw et al. [55] presented a real-time, multiple-camera 3D tracking system to track flying insects. Their method used an extended Kalman filter and the nearest neighbor association to match insects. One of their setups consisted of eleven inexpensive, standard-definition cameras and nine computers for processing camera outputs. The processing speed of their implementation was 60 fps. Other research [56][57] has used Trackit 3D (Biobserve GmbH, Bonn, Germany), a commercial system that uses a stereo camera system to track flying insects in real-time (50 Hz).

1.4 Organization

The rest of this dissertation is organized as follows. Chapter 2 provides a detailed discussion on background information that includes the mathematics, algorithms and methods in photogrammetry and computer vision used in this dissertation research. Chapter 3 provides an introduction to and discussion of the proposed multiple static camera system for real-time object detection, tracking, and 3D positioning system. Chapter 4 provides a discussion of the proposed ‘motion from geometry’ method that includes the ‘collinearity condition of motion’ and ‘tracking from geometry’ methods in detail. Chapter 5 concludes the dissertation and discusses future work.
2. BACKGROUND

This chapter discusses background information to aid in understanding the concepts, algorithms, and methods discussed in this dissertation. The background information discussed here includes the Gauss-Markov model, which is the underlying mathematical model used to solve many of the topics in this dissertation; algorithms and methods in photogrammetry including collinearity condition, space intersection, space resection, and bundle adjustment; and object tracking methods explaining the optical flow constraint and the Lucas-Kanade feature tracker.

2.1 The Gauss-Markov Model

The regular model of observation equations, which is known as the Gauss-Markov model (GMM), is written as equation (2.1) as shown in [58][59]. The linearized form of the observation equations should be used for this model if the equations are non-linear.

\[
y = A \xi + e, \quad e \sim (0, \sigma_o^2 P^{-1}), \quad \text{rk}(A) =: q \leq \{m, n\}
\]  

(2.1)

Here, \(A\) is the \(n \times m\) non-random design (or coefficient) matrix, \(\xi\) is the \(m \times 1\) non-random parameter matrix to be estimated, \(y\) is a \(n \times 1\) random vector of observation, \(e\) is the \(n \times 1\)
vector of observation error, $\sigma_0^2$ is the a priori reference variance, $P$ is the $n \times n$ positive-definite weight matrix, $m$ is the number of observations, and $n$ is the number of parameters. The letter $q$ represents the rank of the design matrix. The degree of freedom (redundancy), $r$, of the equations is defined in equation (2.2).

$$r := n - \text{rank}(A) = n - q$$  

(2.2)

2.1.1 The LESS (Least-Squares Solution) for the GMM

To minimize the quadratic form $e^T P e$ (sum of weighted residuals) subject to the observation equation, the Least-Squares Solution (LESS) is used. The Lagrange target function to minimize is written as:

$$\Phi(\xi) := (y - A \xi)^T P (y - A \xi)$$ is stationary with respect to $\xi$  

(2.3)

The Euler-Lagrange necessary condition and the sufficiency condition are written as equation (2.4).

$$\frac{1}{2} \frac{\partial \Phi}{\partial \xi} = (A^T PA) \hat{\xi} - A^T Py = N \hat{\xi} - c = 0$$  

(2.4)

$$\frac{1}{2} \frac{\partial^2 \Phi}{\partial e \partial e^T} = P$$
From the Euler-Lagrange necessary condition, the solution of the parameter vector is written as:

$$\hat{\xi} = N^{-1}c$$ \hspace{1cm} (2.5)

and the predicted error (i.e., residual) vector is written as:

$$\bar{e} = y - A\hat{\xi}$$ \hspace{1cm} (2.6)

The dispersion of the parameter vector is written as:

$$D\{\hat{\xi}\} = \sigma_o^2 N^{-1}$$ \hspace{1cm} (2.7)

and the estimated reference variance is written as:

$$\hat{\sigma}_o^2 := \frac{\bar{e}^T P \bar{e}}{n - \text{rk}(A)}$$ \hspace{1cm} (2.8)

In the photogrammetry processes used in this dissertation research, the observation matrix \((y)\) consists of the difference between photo coordinate observations and results of collinearity equations in \([mm]\), the design matrix \((A)\) consists of partial derivatives of the collinearity condition equations shown in equation (2.12), and the weight matrix \((P)\) consists of the weight of photo coordinate observations. After adjustment, the estimated
parameter matrix \( (\hat{\xi}) \) shows estimated unknown parameters, the dispersion of the parameter vector represents the covariance matrix, which consists of variances and covariances of estimated parameters, and the predicted error vector \( (\tilde{e}) \) shows the differences between observations and estimated results.

### 2.2 Photogrammetry

The term photogrammetry can be defined as follows: “Photogrammetry is the science of obtaining reliable information about the properties of surfaces and objects without physical contact with the objects, and of measuring and interpreting this information” [60]. Simply put, photogrammetry, as its name implies (i.e., measurements from photographs), is the process that estimates metric information from photographs. The essential task of photogrammetry is to establish the geometric relationship between an object and its representation in the image [61]. Photogrammetry can provide very precise metric solutions from images once the geometric relationship is precisely recovered. In this section, photogrammetry topics related to the concepts used in this dissertation research will be discussed.

#### 2.2.1 Collinearity Condition Equations

The collinearity condition equations or, more simply, collinearity equations, are the most widely used as well as representing the most important concept in photogrammetry. The collinearity condition, as illustrated in Figure 2.1, represents a geometric relationship
wherein a point in the object space \((A)\), its image on a photograph \((a)\), and the perspective center of the photograph \((L)\) are on a straight line in three-dimensional space [62].

![Collinearity condition diagram](image)

Figure 2.1. Collinearity condition; a point in object space \((A)\), its image on a photograph \((a)\), and the perspective center of the photograph \((L)\) are on a straight line in three-dimensional space.

This condition can be expressed by the two expressions in equation (2.9).

\[
x_a = x_o - f \left[ \frac{m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \right] = x_o - f \frac{r}{q}
\]

\[
y_a = y_o - f \left[ \frac{m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \right] = y_o - f \frac{s}{q}
\]

(2.9)

In equation (2.9), \(x_a\) and \(y_a\) are the photo coordinates of an image point; \(x_o\) and \(y_o\) are coordinates of the principal point; \(f\) is the camera focal length; \(X_L, Y_L,\) and \(Z_L\) are the object space coordinates of the exposure station; \(X_A, Y_A,\) and \(Z_A\) are the object space coordinates of a point; and the \(m\)’s are functions of the three rotation angles \((\omega, \varphi,\) and \(\kappa)\) such that:

18
\begin{align*}
m_{11} & = \cos \varphi \cos \kappa \\
m_{12} & = \sin \omega \sin \varphi \cos \kappa + \cos \omega \sin \kappa \\
m_{13} & = -\cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa \\
m_{21} & = -\cos \varphi \sin \kappa \\
m_{22} & = -\sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa \\
m_{23} & = \cos \omega \sin \varphi \sin \kappa + \sin \omega \cos \kappa \\
m_{31} & = \sin \varphi \\
m_{32} & = -\sin \omega \cos \varphi \\
m_{33} & = \cos \omega \cos \varphi 
\end{align*}

Since the collinearity equations are nonlinear, the linearized forms of the equations are used to iteratively solve many photogrammetric solutions. The collinearity condition equations can be linearized by using Taylor’s theorem. In this manner, the collinearity equations in equation (2.9) can be written as equation (2.10) as shown in [62].

\begin{equation}
F = x_o - f \frac{r}{q} = x_a
\end{equation}

\begin{equation}
G = y_o - f \frac{s}{q} = y_a
\end{equation}

The linearized form of equation (2.10) is expressed by taking partial derivatives of the equations with respect to the unknowns (\(\omega, \varphi, \kappa, X_L, Y_L, Z_L, X_A, Y_A, \) and \(Z_A\)). The equations in equation (2.11) show the linearized forms of equation (2.10).
\[ F_0 + \left( \frac{\partial F}{\partial \omega} \right)_0 d\omega + \left( \frac{\partial F}{\partial \phi} \right)_0 d\phi + \left( \frac{\partial F}{\partial \kappa} \right)_0 d\kappa + \left( \frac{\partial F}{\partial X} \right)_0 dX + \left( \frac{\partial F}{\partial Y} \right)_0 dY \]
\[ + \left( \frac{\partial F}{\partial Z} \right)_0 dZ + \left( \frac{\partial F}{\partial X_A} \right)_0 dX_A + \left( \frac{\partial F}{\partial Y_A} \right)_0 dY_A + \left( \frac{\partial F}{\partial Z_A} \right)_0 dZ_A = x_a \]
\[ \text{(2.11)} \]

\[ G_0 + \left( \frac{\partial G}{\partial \omega} \right)_0 d\omega + \left( \frac{\partial G}{\partial \phi} \right)_0 d\phi + \left( \frac{\partial G}{\partial \kappa} \right)_0 d\kappa + \left( \frac{\partial G}{\partial X} \right)_0 dX + \left( \frac{\partial G}{\partial Y} \right)_0 dY \]
\[ + \left( \frac{\partial G}{\partial Z} \right)_0 dZ + \left( \frac{\partial G}{\partial X_A} \right)_0 dX_A + \left( \frac{\partial G}{\partial Y_A} \right)_0 dY_A + \left( \frac{\partial G}{\partial Z_A} \right)_0 dZ_A = y_a \]

Equation (2.11) can be simplified as equation (2.12):

\[ J = b_{11} d\omega + b_{12} d\phi + b_{13} d\kappa - b_{14} dX - b_{15} dY - b_{16} dZ + e_x \]
\[ + b_{17} dX_A + b_{18} dY_A + b_{19} dZ_A + e_{x_A} \]
\[ K = b_{21} d\omega + b_{22} d\phi + b_{23} d\kappa - b_{24} dX - b_{25} dY - b_{26} dZ + e_x \]
\[ + b_{27} dX_A + b_{28} dY_A + b_{29} dZ_A + e_{x_A} \]
\[ \text{(2.12)} \]

where,

- \(e_x\) and \(e_{x_A}\) are observation errors,
- \(\Delta X = X_A - X\)
- \(\Delta Y = Y_A - Y\)
- \(\Delta Z = Z_A - Z\)
- \(b_{11} = \frac{f}{q^2} [r(-m_{33} \Delta Y + m_{32} \Delta Z) - q(-m_{13} \Delta Y + m_{12} \Delta Z)]\)
- \(b_{12} = \frac{f}{q^2} [r(\cos \phi \Delta X + \sin \phi \Delta Y - \cos \phi \Delta Z)]\)
- \(b_{13} = \frac{-f}{2} (m_{11} \Delta X + m_{22} \Delta Y + m_{23} \Delta Z)\)
- \(b_{14} = \frac{f}{q^2} (r m_{31} - q m_{11})\)
- \(b_{15} = \frac{f}{q^2} (r m_{32} - q m_{12})\)
- \(b_{16} = \frac{f}{q^2} (r m_{33} - q m_{13})\)

20
Equation (2.12) is the simplified form of the linearized collinearity equations as shown in [62]. These linearized equations are used to solve many photogrammetric solutions.

### 2.2.2 Interior Orientation

Interior orientation, or the sensor model, is the process that mathematically reconstructs the geometry inside of the camera at the time when a photograph was acquired. The elements of interior orientation include the focal length \( f \) or principal distance, the principal point location \( (x_o, y_o) \), and the parameters for both radial lens distortion \( (k_1, k_2, \text{ and } k_3) \) and for decentering lens distortion \( (p_1, p_2, \text{ and } p_3) \). These elements of interior orientation (i.e., interior orientation parameters, or IOP) can be estimated by the method described in Section 2.2.4. Usually, the coordinates of fiducial marks are included in interior orientation when photographs have fiducial marks. However, the images used
here did not have fiducial marks; therefore fiducial mark coordinates were not included in
the IOP.

2.2.3 Exterior Orientation

Exterior orientation is the process that determines position and attitude of a photograph in
three-dimensional space. The elements of exterior orientation (i.e., exterior orientation
parameters, or EOP) are the spatial position \((X_L, Y_L, \text{ and } Z_L)\) and the angular orientation
\((\omega, \varphi, \text{ and } \kappa)\) of a photograph. The EOP can be estimated by space resection or by using
bundle adjustment techniques.

Figure 2.2. Exterior orientation parameters (EOP) represent position and orientation of
the sensor (camera) in 3D object space.

Figure 2.2 illustrates the exterior orientation parameters. \(XYZ\) represents the three axes of
the three-dimensional object coordinate system while \(xyz\) represents those of the sensor
(camera) coordinate system. The origin of the sensor coordinate system, which is the
perspective center of the camera, is located at \((X_L, Y_L, Z_L)\). Three rotation angles \((\omega, \varphi, \kappa)\)
and κ) represent the rotation angles of the sensor coordinate system with respect to the object coordinate system.

### 2.2.4 Camera Calibration

Here, camera calibration is the process that estimates interior orientation parameters to provide precise metric information via photogrammetric processes. The interior orientation parameters that can be estimated though camera calibration are calibrated focal length (f), symmetric radial (k₁, k₂, and k₃) and decentering (p₁, p₂, and p₃) lens distortions, and location of the principal point (xₒ and yₒ). An analytical self-calibration method can be used to estimate these parameters. Analytical self-calibration is the process that estimates the calibration parameters and photogrammetric solution simultaneously by including the parameter in the solution. Typical collinearity condition equations with camera calibration parameters are written as equation (2.13) as shown in [62]:

\[
x_a = x_o - \bar{x}_a (k_1 r_a^2 + k_2 r_a^4 + k_3 r_a^6) - (1 + p_3^2 r_a^2) \left[ p_1 (3\bar{x}_a^2 + \bar{y}_a^2) + 2p_2 \bar{x}_a \bar{y}_a \right] - f \frac{r}{q}
\]

\[
y_a = y_o - \bar{y}_a (k_1 r_a^2 + k_2 r_a^4 + k_3 r_a^6) - (1 + p_3^2 r_a^2) \left[ p_2 (\bar{x}_a^2 + 3\bar{y}_a^2) + 2p_1 \bar{x}_a \bar{y}_a \right] - f \frac{s}{q}
\]

where,

- \( x_a, y_a \) are measured photo coordinates
- \( x_o, y_o \) are coordinates of the principal point
\[ \bar{x}_a = x_a - x_0 \]
\[ \bar{y}_a = y_a - y_0 \]
\[ r_a^2 = \bar{x}_a^2 + \bar{y}_a^2 \]

- \( k_1, k_2, k_3 \) are symmetric radial lens distortion coefficients
- \( p_1, p_2, p_3 \) are decentering distortion coefficient
- \( f \) is calibrated focal length

Figure 2.3. Lens distortion patterns: symmetric radial (left), decentering (right), and combined symmetric radial and decentering (bottom); distortion magnitudes are exaggerated for visualization. Table 2.1 shows interior orientation parameters for these patterns.
Figure 2.3 illustrates lens distortion patterns. The red crosshairs represent the locations of the principal points, the green rectangles show image boundaries, and the blue arrows represent distortion patterns. The symmetric radial lens distortion (left) shows a distortion pattern along a radial direction from the principal point. The decentering radial lens distortion (right) is caused by imperfect alignment of the lens system and can be divided into asymmetric radial and tangential lens distortion [62]. The interior orientation and other parameters used for Figure 2.3 are listed in Table 2.1. The magnitudes of the lens distortion parameters are exaggerated for visualization.

<table>
<thead>
<tr>
<th>IOP</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>4.6224 mm</td>
</tr>
<tr>
<td>$x_0$</td>
<td>8.9036e-3 mm</td>
</tr>
<tr>
<td>$y_0$</td>
<td>2.1558e-2 mm</td>
</tr>
<tr>
<td>$k_1$</td>
<td>1.5694e-2</td>
</tr>
<tr>
<td>$k_2$</td>
<td>2.0851e-4</td>
</tr>
<tr>
<td>$k_3$</td>
<td>9.9868e-6</td>
</tr>
<tr>
<td>$p_1$</td>
<td>−3.4110e-3</td>
</tr>
<tr>
<td>$p_2$</td>
<td>−9.7042e-5</td>
</tr>
<tr>
<td>Image Size</td>
<td>1292×964</td>
</tr>
<tr>
<td>Pixel Size</td>
<td>0.00375 mm</td>
</tr>
</tbody>
</table>

Table 2.1. Interior orientation parameters used for Figure 2.3

The calibration parameters can have correlation, which can yield poor solution, with exterior orientation parameters when the self-calibration method is used. Therefore, multiple photographs with different positions and orientation should be used to minimize the correlation.
2.2.5 Coordinate Transformations

Since the unit of measurement used in photogrammetry coordinate systems is usually millimeters, a coordinate transformation is needed between the photo coordinate system and the image coordinate system, whose unit is the pixel. Lens distortion also should be considered.

![Diagram of Coordinate Transformations](image)

Figure 2.4. Coordinate transformations between the photo coordinate system and the image coordinate system

Figure 2.4 illustrates the transformation between the photo coordinate system and the image coordinate system. The image space coordinates \((r, c)\) represent row and column, image size \((nRows, nCols)\) represents the size of an image (height, width), and PS represents the pixel size in \([mm/pixel]\). The photo coordinate system is divided into a calibrated photo coordinate system and an uncalibrated photo coordinate system for convenience. Many other literatures use \((x, y)\) for observations in the image coordinate
system instead of \((r, c)\), but here \((r, c)\) is used to avoid confusion with observations in the photo coordinate system. The coordinate transformation from image to uncalibrated photo coordinate system needs image size \((nRows, nCols)\) and pixel size \((mm/pixel)\). It is written as:

\[
x' = \left( c - \frac{nCols}{2} \right) \times PS \quad y' = \left( \frac{nRows}{2} - r \right) \times PS
\]

The inverse transformation of equation (2.14) is written as equation (2.15). Note that \((x', y')\) still needs to consider lens distortion prior to using it in a photogrammetric solution.

\[
c = \frac{x'}{PS} + \frac{nCols}{2} \quad r = \frac{nRows}{2} - \frac{y'}{PS}
\]

The coordinate transformation between uncalibrated and calibrated photo coordinates needs lens-distortion parameters \((k_1, k_2, k_3, p_1, p_2, \text{ and } p_3)\), which can be estimated using the self-calibration method. The coordinate transformation from uncalibrated photo coordinates to the calibrated photo coordinates is written as equation (2.16) [63][62].

\[
x_a = x_a' + x_a \left( k_1 r_a^2 + k_2 r_a^4 + k_3 r_a^6 \right) + \left( 1 + p_2^2 r_a^2 \right) \left[ p_1 \left( 3 \tilde{x}_a^2 + \tilde{y}_a^2 \right) + 2 p_2 \tilde{x}_a \tilde{y}_a \right]
\]

\[
y_a = y_a' + y_a \left( k_1 r_a^2 + k_2 r_a^4 + k_3 r_a^6 \right) + \left( 1 + p_2^2 r_a^2 \right) \left[ p_2 \left( \tilde{x}_a^2 + 3 \tilde{y}_a^2 \right) + 2 p_1 \tilde{x}_a \tilde{y}_a \right]
\]

27
The coordinate transformation from object coordinates to calibrated photo coordinates needs the collinearity condition equations (2.9). The inverse transformation cannot be calculated directly. However, it can be calculated in an iterative manner. The inverse transformation needs photo coordinates observations from two different cameras: this process is called space intersection. To simplify the equation, the lens distortion terms are defined in equation (2.17):

\[
x_r = \bar{x}_a (k_1 r_a^2 + k_2 r_a^4 + k_3 r_a^6)
\]
\[
x_a = (1 + p_2^2 r_a^2) [p_1 (3\bar{x}_a^2 + \bar{y}_a^2) + 2 p_2 \bar{x}_a \bar{y}_a]
\]
\[
y_r = \bar{y}_a (k_1 r_a^2 + k_2 r_a^4 + k_3 r_a^6)
\]
\[
y_d = (1 + p_2^2 r_a^2) [p_2 (3\bar{x}_a^2 + \bar{y}_a^2) + 2 p_2 \bar{x}_a \bar{y}_a]
\]

where,

- \(x_r\) is radial lens distortion term for \(x\) axis
- \(x_d\) is decentering lens distortion term for \(x\) axis
- \(y_r\) is radial lens distortion term for \(y\) axis
- \(y_d\) is decentering lens distortion term for \(y\) axis

Then the collinearity condition equations are written as equation (2.18).

\[
x_{o \prime} + x_r + x_d = x_o = \frac{f_r}{q} \]
\[
y_{o \prime} + y_r + y_d = y_o = \frac{f_s}{q}
\]
Equations (2.18) show the relationship between calibrated and uncalibrated photo coordinates.

### 2.2.6 Space Intersection

Space intersection is the photogrammetric process that estimates the 3D coordinates of a point in the object space from photo coordinates observations in a stereo pair.

![Conceptual diagram of space intersection: the space intersection method estimates 3D coordinates of a point in the object space form photo coordinates observations in a stereo pair.](image)

Figure 2.5. Conceptual diagram of space intersection: the space intersection method estimates 3D coordinates of a point in the object space from photo coordinates observations in a stereo pair.

Figure 2.5 illustrates the concept of space intersection. Known values are the exterior orientation parameters of two photographs \((L_1, L_2)\), photo coordinates of a point \((a_1, a_2)\) in each photograph, and the initial value for the point in object space \(A\).
Equation (2.19) shows the linearized form of the collinearity equations for space intersection. Since the exterior orientation parameters of each photograph are known, the object coordinate terms \((dX_A, dY_A, dZ_A)\) are estimated iteratively in the linearized form of the collinearity condition equation (2.12).

\[
J = b_{14} dX_A + b_{15} dY_A + b_{16} dZ_A + e_{ix}
\]

\[
K = b_{24} dX_A + b_{25} dY_A + b_{26} dZ_A + e_{iy}
\]

(2.19)

Equation (2.20) shows the matrix form of equation (2.19). Subscripts \(C_0\) and \(C_1\) represent cameras 0 and 1 (or photographs 0 and 1), respectively. The solution of equation (2.20) can be estimated iteratively using LESS as discussed in Section 2.1.1. The sizes of the design matrix and the observation matrix are \(4 \times 3\) and \(4 \times 1\), respectively. Table 2.2 shows the iterative space intersection algorithm.

\[
\begin{bmatrix}
(J)_{C_0} \\
(K)_{C_0} \\
(J)_{C_1} \\
(K)_{C_1}
\end{bmatrix} =
\begin{bmatrix}
(b_{14})_{C_0} & (b_{15})_{C_0} & (b_{16})_{C_0} \\
(b_{24})_{C_0} & (b_{25})_{C_0} & (b_{26})_{C_0} \\
(b_{14})_{C_1} & (b_{15})_{C_1} & (b_{16})_{C_1} \\
(b_{24})_{C_1} & (b_{25})_{C_1} & (b_{26})_{C_1}
\end{bmatrix}
\begin{bmatrix}
dX_A \\
dY_A \\
dZ_A
\end{bmatrix} + e
\]

(2.20)
**Data:** photo coordinate observations: $(x_a, y_a)_{C_0}, (x_a, y_a)_{C_1}$

**Initial values of point $A$:** $(X_A, Y_A, Z_A)$

**While** ($\|\varepsilon\|_{\varepsilon} > \text{threshold}$)

**Calculate design and $y$ matrix:**

$$A = \begin{bmatrix}
(b_{14})_{C_0} & (b_{15})_{C_0} & (b_{16})_{C_0} \\
(b_{24})_{C_0} & (b_{25})_{C_0} & (b_{26})_{C_0} \\
(b_{14})_{C_1} & (b_{15})_{C_1} & (b_{16})_{C_1} \\
(b_{24})_{C_1} & (b_{25})_{C_1} & (b_{26})_{C_1}
\end{bmatrix}$$

$$y = \begin{bmatrix}
(J)_{C_0} \\
(K)_{C_0} \\
(J)_{C_1} \\
(K)_{C_1}
\end{bmatrix}$$

**Estimate parameter matrix:**

$$\hat{\varepsilon} = \begin{bmatrix}
dX_A \\
dY_A \\
dZ_A
\end{bmatrix} = (A^T A)^{-1} A^T y$$

**Update solution:**

$$X_A \quad Y_A \quad Z_A = \begin{bmatrix}
X_A \\
Y_A \\
Z_A
\end{bmatrix} + \begin{bmatrix}
dX_A \\
dY_A \\
dZ_A
\end{bmatrix}$$

**End**

**Result:** 3D object space coordinates of point $A$

---

Table 2.2. Space intersection algorithm
2.2.7 Space Intersection with Multiple Cameras

The bundle adjustment technique can be used to solve for the space intersection solution in the multiple camera case. Figure 2.6 illustrates the concept of space intersection for the multiple camera case. Known values are the exterior orientation parameters of the cameras or photographs ($L_1$, $L_2$, $L_3$, and $L_4$), the photo coordinates of a point ($a_1$, $a_2$, $a_3$, and $a_4$), and the initial value for the point in object space ($A$).

\[
\begin{bmatrix}
(J)_{C_0} \\
(K)_{C_1} \\
(J)_{C_2} \\
(K)_{C_3} \\
\end{bmatrix} \begin{bmatrix}
(b_{14})_{C_0} & (b_{15})_{C_0} & (b_{16})_{C_0} \\
(b_{14})_{C_1} & (b_{15})_{C_1} & (b_{16})_{C_1} \\
(b_{14})_{C_2} & (b_{15})_{C_2} & (b_{16})_{C_2} \\
(b_{14})_{C_3} & (b_{15})_{C_3} & (b_{16})_{C_3} \\
\end{bmatrix}
\begin{bmatrix}
{dX_a} \\
{dY_a} \\
{dZ_a} \\
\end{bmatrix} + e
\]  

(2.21)

Equation (2.21) shows matrix form of equation (2.19) for the multiple camera case. In this equation, subscripts $C_0$, $C_1$, $C_2$ and $C_3$ represent cameras 0, 1, 2, and 3, respectively. Solution of equation (2.21) can be estimated iteratively by using LESS, as discussed in Section 2.1.1. The sizes of the design matrix and the observation matrix are $2n\times3$ and $2n\times1$, respectively; where $n$ is the number of cameras (or photographs).
2.2.8 Space Resection

Space resection is the photogrammetric procedure that estimates the exterior orientation parameters \( (X_L, Y_L, Z_L, \omega, \varphi, \kappa) \) of a photograph using at least three control points with known \( XYZ \) coordinates [64]. A linearized form of the collinearity condition equations is used to solve for the exterior orientation parameters.

Figure 2.7. Conceptual diagram of space resection: the space resection method estimates the exterior orientation parameters of a photograph using known control points.
Figure 2.7 illustrates the concept of space resection. Space resection estimates the six unknown exterior orientation parameters ($\omega$, $\phi$, $\kappa$, $X_L$, $Y_L$, and $Z_L$) using a linearized form of the collinearity condition equations with known control points. Each control point ($x$, $y$) provides two equations. Therefore, at least three control points are needed to estimate the six unknown exterior orientation parameters.

$$J = b_1 d\omega + b_{12} d\phi + b_{13} d\kappa - b_{14} dX_L - b_{15} dY_L - b_{16} dZ_L + e_{x_0}$$

$$K = b_{21} d\omega + b_{22} d\phi + b_{23} d\kappa - b_{24} dX_L - b_{25} dY_L - b_{26} dZ_L + e_{y_0}$$

(2.22)

Equation (2.22) shows the linearized form of the space resection equations. Since the object coordinates of each control point are known, only the exterior orientation parameter terms ($d\omega$, $d\phi$, $d\kappa$, $dX_L$, $dY_L$, and $dZ_L$) are unknown and estimated in the linearized collinearity condition equations (2.12).

$$\begin{bmatrix}
(J)_d \\
(K)_d \\
(J)_g \\
(K)_g \\
(J)_c \\
(K)_c \\
\end{bmatrix} = 
\begin{bmatrix}
(b_{11})_d & (b_{12})_d & (b_{13})_d & -(b_{14})_d & -(b_{15})_d & -(b_{16})_d \\
(b_{21})_d & (b_{22})_d & (b_{23})_d & -(b_{24})_d & -(b_{25})_d & -(b_{26})_d \\
(b_{11})_g & (b_{12})_g & (b_{13})_g & -(b_{14})_g & -(b_{15})_g & -(b_{16})_g \\
(b_{21})_g & (b_{22})_g & (b_{23})_g & -(b_{24})_g & -(b_{25})_g & -(b_{26})_g \\
(b_{11})_c & (b_{12})_c & (b_{13})_c & -(b_{14})_c & -(b_{15})_c & -(b_{16})_c \\
(b_{21})_c & (b_{22})_c & (b_{23})_c & -(b_{24})_c & -(b_{25})_c & -(b_{26})_c \\
\end{bmatrix} 
\begin{bmatrix}
d\omega \\
d\phi \\
d\kappa \\
dX_L \\
dY_L \\
dZ_L \\
\end{bmatrix} + e$$

(2.23)
Equation (2.23) shows the matrix form of equation (2.22). Subscripts $A$, $B$ and $C$ represent control points $A$, $B$, and $C$, respectively. The solution of equation (2.23) can be estimated iteratively using LESS, as discussed in Section 2.1.1. The sizes of the design matrix and the observation matrix are $2n \times 6$ and $2n \times 1$, respectively, where $n$ is the number of control points.

### 2.2.9 Bundle Adjustment

Bundle adjustment is the photogrammetric process that adjusts all photogrammetric observations to control points in a single solution [62].

![Figure 2.8. Conceptual diagram of bundle adjustment: the bundle adjustment estimates all photogrammetric observations simultaneously using control points.](image)

Figure 2.8 illustrates the concept of bundle adjustment. The bundle adjustment estimates all photogrammetric observations including the exterior orientation parameters ($\omega$, $\varphi$, $\kappa$, $X_L$, $Y_L$, and $Z_L$) of all photographs ($L_1$, $L_2$), and the object space coordinates ($X$, $Y$, and $Z$) of all object points ($A$, $B$, and $C$). The number of unknowns for the bundle adjustment is 35.
\((i \times 6) + (j \times 3)\), where \(i\) is the number of photographs and \(j\) is the number of unknown object coordinates. The equation for bundle adjustment is written simply in equation (2.24) as shown in [62]:

$$
\tilde{B}_y \hat{\Delta}_i + \tilde{B}_y \hat{\Delta}_j = \varepsilon_y
$$

(2.24)

where,

$$
\tilde{B}_y = \begin{bmatrix}
    b_{1i} & b_{12i} & b_{13i} & \cdots & b_{14i} & b_{15i} & b_{16i} \\
    b_{2i} & b_{22i} & b_{23i} & \cdots & b_{24i} & b_{25i} & b_{26i} \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    b_{6i} & b_{26i} & b_{36i} & \cdots & b_{66i} & b_{76i} & b_{86i}
\end{bmatrix}
$$

$$
\hat{\Delta}_i = \begin{bmatrix}
    \frac{d\omega_i}{dX_i} \\
    \frac{d\phi_i}{dX_i} \\
    \frac{dk_i}{dX_i} \\
    \frac{dX_i}{dY_i} \\
    \frac{dY_i}{dZ_i} \\
    \frac{dZ_i}{dZ_i}
\end{bmatrix}
$$

$$
\hat{\Delta}_j = \begin{bmatrix}
    \frac{dX_j}{dX_j} \\
    \frac{dY_j}{dY_j} \\
    \frac{dZ_j}{dZ_j}
\end{bmatrix}
$$

$$
\varepsilon_y = \begin{bmatrix}
    J_y \\
    K_y
\end{bmatrix}
$$

In equation (2.24), \(\hat{\Delta}_i\) gives the corrections to the exterior orientation parameters of photograph \(i\), \(\hat{\Delta}_j\) gives the corrections to the object space coordinates of point \(j\), \(\tilde{B}_y\) and \(\tilde{B}_y\) are partial derivatives of the collinearity condition equations with respect to each of the six exterior orientation parameters of photograph \(i\) and the three object space coordinates of point \(j\), respectively; and \(\varepsilon_y\) contains measured minus estimated photo coordinates for point \(j\) on photograph \(i\).
2.2.10 Epipolar Geometry

In stereo camera geometry, the epipolar plane is the plane containing a baseline and an arbitrary object point [62][65]. The epipolar line is the intersection line of an epipolar plane with an image plane. Epipolar geometry is used to find conjugate image points in stereo matching [65] by reducing a two-dimensional search to a one-dimensional search along with the epipolar line [61].

Figure 2.9 illustrates epipolar geometry. An epipolar plane is a plane that is defined by three points (the perspective centers of the two cameras and a point in the object space: \( L_1, L_2, \) and \( A \)). The epipolar plane always contains a baseline (i.e., the line that connects the perspective centers of two cameras; \( L_1 \) and \( L_2 \)). A line passing through \( a_1 \) and \( e_1 \) on an image plane is called an epipolar line. Referring to Figure 2.9, this corresponds to point \( A \) in 3D object space or point \( a_2 \) in 2D image space. An epipolar line is also the intersection line between an epipolar plane and the image plane. The epipole (\( e_1 \) or \( e_2 \)) is the
intersection point between the baseline and the image plane. All epipolar lines on an image intersect at the epipole.

2.3 Object Tracking

In this section, object tracking and image matching methods are discussed. The object tracking method used in this dissertation is the Lucas-Kanade feature tracker [66]. The Lucas-Kanade method and the optical flow constraint equation, which is the underlying base concept for the Lucas-Kanade method, are here discussed. Template matching with cross-correlation method is also discussed.

2.3.1 Optical Flow Constraint Equation

The optical flow constraint equation is based on the concept that intensity values of images are the same with respect to time. Assuming that $I(x, y, t)$ represents an intensity value of a point at $(x, y)$ in an image at time $t$, then the intensity value remains the same after small changes in location $(dx, dy)$ and time $(dt)$. This assumption is written as equation (2.25) as shown in [6][67].

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$  \hspace{1cm} (2.25)
Taking partial derivatives of the right-hand side of the equation (2.25) with respect to the unknowns \((dx, dy, \text{ and } dt)\) and neglecting high-order terms provides equation (2.26).

\[
I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt
\]  

(2.26)

Substituting equation (2.25) into equation (2.26) provides equation (2.27).

\[
\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0
\]  

(2.27)

Dividing equation (2.27) by \(dt\) provides equation (2.28).

\[
\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0
\]  

(2.28)

In equation (2.28), \(dx/dt\) and \(dy/dt\) are motions in the \(x\) and \(y\) directions, respectively, as shown in equation (2.29). Assuming \(u\) and \(v\) are motions in the \(x\) and \(y\) directions, respectively, then:

\[
u = \frac{dy}{dt}
\]  

(2.29)
Therefore, equation (2.28) can be written as equation (2.30).

\[
\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v = -\frac{\partial I}{\partial t} \tag{2.30}
\]

Equation (2.30) is called the optical flow constraint equation. This equation can be written alternatively as equation (2.31).

\[
I_x u + I_y v = -I_t \tag{2.31}
\]

In equation (2.31), \(I_x\) and \(I_y\) represent the spatial rates of change of intensity in the \(x\) and \(y\) directions, respectively. \(I_t\) represents the change of intensity with respect to time. The optical flow constraint equation has two unknowns (\(u\) and \(v\)); therefore, it needs at least two equations to solve for the unknown parameters. The optical flow constraint equation represents a line in the \((u, v)\) coordinate system, as shown in Figure 2.10.

Figure 2.10. The optical flow constraint equation
2.3.2 Lucas-Kanade Feature Tracker

The Lucas-Kanade algorithm introduced in [66] is one of the most widely used tracking methods in the computer vision field [68]. The Lucas-Kanade feature tracker assumes that the motion of neighboring pixels is the same and very small. Therefore, the optical flow constraint equation for neighboring pixels can be used to calculate the motion and the image pyramid method can be used to calculate larger motion. Bouguet [69][70] explains the Lucas-Kanade method in detail.

\[ I_x u + I_y v = -I_t \]  \hspace{1cm} (2.32)

Equation (2.32) shows the optical flow constraint equation for one pixel in two consecutive images. Assuming that \( A \) and \( B \) represent intensity values of an image at times \( t \) and \( t+dt \), respectively, then \( I_x, I_y, \) and \( I_t \) can be calculated by equation (2.33).

\[
I_x(x,y) = \frac{\partial A(x,y)}{\partial x} = \frac{A(x+1,y) - A(x-1,y)}{2}
\]

\[
I_y(x,y) = \frac{\partial A(x,y)}{\partial y} = \frac{A(x,y+1) - A(x,y-1)}{2}
\]

\[
I_t(x,y) = \frac{\partial A(x,y)}{\partial t} = \frac{A(x,y) - B(x,y)}{2}
\]  \hspace{1cm} (2.33)
The optical flow constraint equations for a small area that consists of neighboring pixels can be written as equation (2.34).

\[
\begin{bmatrix}
  I_x & I_y \\
  I_x & I_y \\
  \vdots & \vdots \\
  I_x & I_y \\
\end{bmatrix}
\begin{bmatrix}
  u \\
  v \\
\end{bmatrix} =
\begin{bmatrix}
  -I_t \\
  -I_t \\
  \vdots \\
  -I_t \\
\end{bmatrix}
\]  

(2.34)

Each row of equation (2.34) represents the optical flow equation for one pixel. The LESS of equation (2.34) is written as equation (2.35).

\[
\begin{bmatrix}
  \sum I_x^2 & \sum I_x I_y \\
  \sum I_x I_y & \sum I_y^2 \\
\end{bmatrix}
\begin{bmatrix}
  u \\
  v \\
\end{bmatrix} =
\begin{bmatrix}
  -\sum I_x I_t \\
  -\sum I_y I_t \\
\end{bmatrix}
\]  

(2.35)

Therefore, the motion is calculated by equation (2.36).

\[
\begin{bmatrix}
  u \\
  v \\
\end{bmatrix} =
\begin{bmatrix}
  \sum I_x^2 & \sum I_x I_y \\
  \sum I_x I_y & \sum I_y^2 \\
\end{bmatrix}^{-1}
\begin{bmatrix}
  -\sum I_x I_t \\
  -\sum I_y I_t \\
\end{bmatrix}
\]  

(2.36)

Practically, it is necessary to use the image pyramid method because the optical flow constraint equation is satisfied only when motion is small. The solution of equation (2.36) is not motion itself but the change of motion. Therefore an iterative method is needed to calculate accurate motion until there is no more change [70].
**Data:** image sequence: \( A(x, y), B(x, y) \)

**Generate image pyramid:** \( A_0(x, y) \cdots A_n(x, y), \quad B_0(x, y) \cdots B_n(x, y) \)

**Initial values of motion:** \( u_0 = 0, \quad v_0 = 0 \)

**For** \( i = n \to 0 \)

- **Calculate** \( I_x \) and \( I_y \):
  \[
  I_x(x, y) = \frac{A_i(x + 1, y) - A_i(x - 1, y)}{2} \\
  I_y(x, y) = \frac{A_i(x, y + 1) - A_i(x, y - 1)}{2}
  \]

- **While** \( ||du + dv|| > \text{threshold} \)

  - **Calculate** \( I_t \):
    \[
    I_t(x, y) = \frac{A_i(x, y) - B_i(x + u_i, y + v_i)}{2}
    \]

  - **Calculate** \( du, dv \):
    \[
    \begin{bmatrix}
    du \\
    dv
    \end{bmatrix} = \begin{bmatrix}
    \sum I_x^2 & \sum I_x I_y \\
    \sum I_x I_y & \sum I_y^2
    \end{bmatrix}^{-1} \begin{bmatrix}
    - \sum I_x I_t \\
    - \sum I_y I_t
    \end{bmatrix}
    \]

  - **Update motion**:
    \[
    \begin{bmatrix}
    u_i \\
    v_i
    \end{bmatrix} = \begin{bmatrix}
    u_i \\
    v_i
    \end{bmatrix} + \begin{bmatrix}
    du \\
    dv
    \end{bmatrix}
    \]

**End**

- **Update motion**:
  \[
  \begin{bmatrix}
  u_{i+1} \\
  v_{i+1}
  \end{bmatrix} = 2 \begin{bmatrix}
  u_i \\
  v_i
  \end{bmatrix}
  \]

**End**

**Result:** image motion \( u \) and \( v \)

Table 2.3. Lucas-Kanade feature tracker algorithm

Table 2.3 shows the Lucas-Kanade feature tracker algorithm. At each pyramid level, \( I_x \) and \( I_y \) are calculated at once and remain as constants throughout the iteration. Only \( I_t \) is updated at every iteration using the change of motion.
2.3.3 Image Matching

In this section, the template matching method with cross correlation is discussed. Template matching finds parts of an image which match a predefined template image [71][72]. This is illustrated in Figure 2.11.

![Image of template matching](image)

Figure 2.11. Template matching finds parts of original image (left) that correspond to a template (middle). The matching result is marked by a red rectangle in the right.

Assuming that there are image patches \( A \) and \( B \) whose sizes are \( n \times m \), then the average intensity values of both patch \( A \) and patch \( B \) can be calculated by equation (2.37).

\[
\overline{g}_A = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} A(x_i, y_j)}{n \cdot m} \\
\overline{g}_B = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} B(x_i, y_j)}{n \cdot m} 
\]  (2.37)
The standard deviations of image patches $A$ and $B$ are written as equation (2.38).

\[
\sigma_A = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (A(x_i, y_j) - \overline{A})^2}{n \cdot m - 1}} \quad \sigma_B = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (B(x_i, y_j) - \overline{B})^2}{n \cdot m - 1}} \tag{2.38}
\]

The covariance of image patches $A$ and $B$ is written as equation (2.39).

\[
\sigma_{AB} = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (A(x_i, y_j) - \overline{A})(B(x_i, y_j) - \overline{B})}{n \cdot m - 1}} \tag{2.39}
\]

The correlation coefficient is defined in equation (2.40).

\[
\rho = \frac{\sigma_{AB}}{\sigma_A \sigma_B} \tag{2.40}
\]

The correlation coefficient, $\rho$, has following characteristic:

\[-1 \leq \rho \leq 1 \tag{2.41}\]

If the correlation coefficient, $\rho$, is equal to 1, it means that the two patches are identical. If the correlation coefficient, $\rho$, is equal to $-1$, this indicates inverse correlation.
Figure 2.12. Conceptual diagram of template matching

Figure 2.12 illustrates concept of template matching. Assuming a patch \( A \) is the template and a patch \( B \) moves in the search space (red rectangle) to calculate and compare the correlation coefficients. Since template matching is a highly time and process consuming task, reducing the size of the search space is very important.

Figure 2.13. Conceptual diagrams of epipolar constraint (left) and vertical line locus constraint (right) for image matching: epipolar constraint reduces search area in a single epipolar line while vertical line locus constraint reduces the search area on the epipolar line.
Figure 2.13 shows conceptual diagrams of the epipolar and the vertical line locus constraints. These constraints are commonly used in photogrammetry to reduce the size of a search space. The epipolar constraint is that a point on an image corresponds to a line on the other image of a stereo pair. The vertical line locus constraint can restrict the epipolar line by using the maximum and the minimum height of a target area. The vertical line locus concept is commonly used to generate a DSM (Digital Surface Model) from an aerial stereo pair [72].

Figure 2.14. Reducing search space with the epipolar constraint in template matching

Figure 2.14 shows the concept of template matching using the epipolar constraint. When a stereo pair is used and the exterior orientation parameters of the stereo pair are known, a search space (red rectangle) can be reduced within an epipolar line (blue line).
Figure 2.15 illustrates the concept of search space reduction using a vertical line locus constraint in a stereo pair. The maximum and minimum height of the target area (distance between cameras and a target object for the terrestrial case) can define the end points of an epipolar line.
In this chapter, a real-time object detection, tracking, and 3D positioning system using a static multiple-camera setup is introduced. The system consists of four Gigabit Ethernet cameras, a network switch, and a processing computer. The processing system is divided into detection, tracking, and refinement stages. In the detection stage, the system finds a target object using an image matching technique without prior information. Tracking finds the object from its position in the previous frame. In the refinement stage, statistics from bundle adjustments are used to refine results from the detection and tracking stages to maintain reliability. Experimental results have shown that this system provides a less than $\pm 5 \, \text{mm}$ average positioning accuracy for a setup consisting of four high-definition cameras with data processing at speeds of six to ten frames per second running on a single laptop computer.

3.1 System Configuration

The developed system consists of a processing computer, four static high-definition Gigabit Ethernet (GigE) cameras, and a Gigabit Ethernet switch (Figure 3.1). The four
cameras are placed on the four corners of the target area to provide better geometry with wide baselines and to reduce occlusion effects.

Figure 3.1. System configuration of multiple static camera real-time object detection, tracking, and 3D positioning system [73]

A wired network was chosen for the four Gigabit Ethernet (GigE) Vision cameras because the data transmittal rate is an important factor for real-time data processing. The Ethernet communication protocol can transmit data up to 1000 Mbit/s over a cable length of up to 100 meters. This 1000 Mbit/s data transmit rate enabled the use of four cameras for real-time processing. The cameras selected for the system provided a resolution of $1292 \times 964$ pixels and frame rates of up to 30 fps (frame per second). The selected camera and lens are shown in Figure 3.2.
3.1.1 Interior Orientation of Cameras

An automated camera calibration program was developed to automatically detect control points and provide estimates of the interior orientation parameters using the algorithms described in Section 2.2.4. The estimated interior orientation parameters for the cameras are listed in Table 3.1. The size of each pixel is 0.0035×0.0035 mm.

<table>
<thead>
<tr>
<th>IOP</th>
<th>CAM0</th>
<th>CAM1</th>
<th>CAM2</th>
<th>CAM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>4.9793 mm</td>
<td>4.9987 mm</td>
<td>4.9712 mm</td>
<td>5.0872 mm</td>
</tr>
<tr>
<td>x₀</td>
<td>3.1983e-2 mm</td>
<td>-1.1360e-1 mm</td>
<td>7.1858e-2 mm</td>
<td>-3.2459e-2 mm</td>
</tr>
<tr>
<td>y₀</td>
<td>-3.8755e-3 mm</td>
<td>5.9029e-2 mm</td>
<td>5.4991e-2 mm</td>
<td>-3.1969e-2 mm</td>
</tr>
<tr>
<td>k₁</td>
<td>2.7250e-3</td>
<td>3.3587e-3</td>
<td>2.4753e-3</td>
<td>2.8160e-3</td>
</tr>
<tr>
<td>k₂</td>
<td>-2.1683e-4</td>
<td>-3.8075e-4</td>
<td>-1.7631e-4</td>
<td>-2.6553e-4</td>
</tr>
<tr>
<td>k₃</td>
<td>4.0233e-6</td>
<td>1.9075e-5</td>
<td>-2.7181e-8</td>
<td>7.0613e-6</td>
</tr>
<tr>
<td>p₁</td>
<td>1.7749e-4</td>
<td>5.5350e-4</td>
<td>-9.3139e-5</td>
<td>2.2295e-4</td>
</tr>
<tr>
<td>p₂</td>
<td>-8.3384e-5</td>
<td>-8.5702e-5</td>
<td>-5.6447e-5</td>
<td>-5.7827e-5</td>
</tr>
</tbody>
</table>

Table 3.1. Estimated interior orientation parameters for cameras used in the system
Figure 3.3 illustrates distribution of camera positions (left), control points (right top) for the camera calibration, and lens distortion patterns of CAM0 (right bottom) estimated by the automatic camera calibration program.

![Figure 3.3. Distribution of camera positions (left) and detected control points (right top) for camera calibration with lens distortion patterns of CAM0 (right bottom)](image)

### 3.1.2 Target and Camera Installation

Eighty-two targets were installed as control points in a room measuring 8×5×2.5 meters and the 3D coordinates of these points were determined using a Total Station. A Total Station is an optical surveying instrument that can precisely measure angle and distance simultaneously. Figure 3.4 illustrates the distribution of the control points in this room.
Figure 3.4. Distribution of 82 control points in an 8×5×2.5 m room that were used to determine the exterior orientation parameters of the cameras in the system.

The cameras were installed in each corner of the room to maximize coverage and reduce occlusion effects. Figure 3.5 shows installed positions of these cameras.

Figure 3.5. Installed cameras: (from left to right) CAM0, CAM1, CAM2, and CAM3
### 3.1.3 Exterior Orientation

The exterior orientation parameters were estimated using the bundle adjustment technique as described in Section 2.2.9. Their values are listed in Table 3.2.

<table>
<thead>
<tr>
<th>EOP</th>
<th>CAM0</th>
<th>CAM1</th>
<th>CAM2</th>
<th>CAM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$ [rad]</td>
<td>4.31663518 ±0.00101515</td>
<td>5.07830736 ±0.00144354</td>
<td>5.17328518 ±0.00145738</td>
<td>4.53375784 ±0.00095651</td>
</tr>
<tr>
<td>$\varphi$ [rad]</td>
<td>-2.14026551 ±0.00061101</td>
<td>-1.09497969 ±0.00112015</td>
<td>1.11721326 ±0.00095071</td>
<td>2.23444908 ±0.00077702</td>
</tr>
<tr>
<td>$\kappa$ [rad]</td>
<td>5.95115813 ±0.00067409</td>
<td>6.60946182 ±0.00122208</td>
<td>-6.7088436 ±0.00129988</td>
<td>-6.07571915 ±0.00057485</td>
</tr>
<tr>
<td>$X_L$ [m]</td>
<td>146.87357 ±0.00132</td>
<td>146.88154 ±0.00244</td>
<td>152.77997 ±0.00242</td>
<td>152.83557 ±0.00216</td>
</tr>
<tr>
<td>$Y_L$ [m]</td>
<td>244.74934 ±0.00444</td>
<td>248.34611 ±0.00774</td>
<td>248.35717 ±0.00677</td>
<td>244.11397 ±0.00539</td>
</tr>
<tr>
<td>$Z_L$ [m]</td>
<td>352.28108 ±0.00311</td>
<td>352.28016 ±0.00428</td>
<td>352.27812 ±0.00402</td>
<td>352.29473 ±0.00379</td>
</tr>
</tbody>
</table>

Table 3.2. Estimated exterior orientation parameters for CAM0, CAM1, CAM2, and CAM3

Figure 3.6 illustrates the positions and orientations of the cameras from the estimated exterior orientation parameters. Note that the sizes of cameras (focal length and image plane) are exaggerated fifty times for visualization.

### 3.1.4 Time Synchronization of Cameras and Frame Rate

Time synchronization between the cameras is very important in estimating the 3D coordinates of a moving object. The software trigger method was used here, a method which sends an image acquisition request to all cameras in the setup at the same time.
epoch and then receives time-synchronized images back from all these cameras. The frame rate is defined based on both image acquisition delay and processing delay. The image acquisition delay strongly depends on the exposure time, which is related to the lighting necessary to acquire a clean image. Therefore, the optimal exposure time that compromises between a good frame rate and acceptable image quality was chosen prior to processing images.

Figure 3.6. Camera exterior orientation results (focal length and size of image plane are exaggerated 50 times for visualization)

3.2 Detection, Tracking and Refinement

Detection is here defined as the process that finds a predefined target object in an image without knowing its position in the previous frame. On the other hand, tracking is defined as the process of finding a predefined target object based on its known position in the
previous frame. Detection finds the target object using the image matching method described in Section 2.3.3 and then verifies the result using epipolar correspondence. Tracking traces the target object using the Lucas-Kanade method described in Section 2.3.2 and then verifies the result using image matching again. If detection is successful in a camera, the camera changes its mode from detection to tracking; otherwise, the camera remains in detection mode. If tracking fails in a camera, the camera changes its mode from tracking to detection; otherwise, the camera remains in tracking mode. Since detection and tracking results do not always provide correct solutions, a refinement step is needed to refine the detection and tracking results. This refinement step uses statistics from bundle adjustments to verify the results from both detection and tracking.

3.2.1 The Target Object

Here, the goal was to find the 3D position of a construction drill (shown in Figure 3.7, left) in real-time. Since the entire body of the tool is difficult to detect because it changes its shape significantly with respect to viewing angle, it was necessary to select a part of the tool that appears as much the same as possible regardless of any change in viewpoint. Therefore a pattern of black, white, and black stripes (BWB) was installed around the chuck of this drill. In Figure 3.7, a yellow box outlines this pattern (BWB). This pattern can be detected easily by using specially designed templates that are scale and rotation invariant (Figure 3.7, right).
To detect this pattern, multiple scaled template sets were generated that consisted of multiple rotated templates. The number of the scale sets can be calculated based on the size of the pattern (BWB) and the maximum and minimum distances of the pattern from the cameras. The angle interval between rotated templates was set to ten degrees.

### 3.2.2 Detection

In this system, detection is defined as the process that finds a pattern without knowing its location in the previous frame. Detection is the most time consuming step in the system if the search space for image matching is large. In order to reduce detection processing time, color segmentation that reduces the search space was used in this experiment. Then the search for the pattern (BWB) was performed using image matching with sets of the scale and rotation invariant templates. The scale of the template was selected from the distance between the cameras and the pattern (BWB) using the 3D coordinates of the pattern (BWB) in the previous frame, if available. Once the scale was selected, template matching was performed for all rotation angles in that scale to find the candidates.
Practically, the image matching result that shows the highest matching score does not always provide a correct result due to many reasons including changes in illumination, occlusion, etc. Therefore, another step is necessary to verify the image matching result. Since multiple cameras were used in this system, the geometric constraints between cameras can be used to find the correct pattern (black-white-black stripes) among the candidates. Figure 3.8 illustrates the epipolar correspondence concept used to find the correct match. The top five BWB candidates, i.e., those which show the top five matching scores from each camera, were selected and then epipolar correspondence, which is a point in a camera that corresponds to a line in the other camera, was used to find possible combinations of matches. For example, in Figure 3.8, green rectangles in the left blue circle represent BWB candidates in CAM0 and those of the right blue circle
represent BWB candidates detected in CAM1. The green rectangle 1 in CAM0 (left) corresponds to red line in CAM1 (right). In this figure, it can be seen that the BWB candidate 1 in CAM0 corresponds to BWB candidates 0 and 3 that are inside the buffered area of the red line in CAM1. The thickness of the buffered area changes with respect to the distance between the camera and the candidate to avoid a possible exclusion. Once all correspondences are established between all cameras and all candidates, a correct BWB can be found that satisfies both high matching score and epipolar correspondences between all cameras. Once the correct BWB is found, the camera changes its mode from detection to tracking in the following frame. Test results have shown that the processing speed is six frames per second when all four cameras are in detection mode.

### 3.2.3 Tracking

Tracking finds position of the target object in an image when its position in the previous frame is known. The Lucas-Kanade method described in Section 2.3.2 was used for this tracking step. Tracking also does not always provide correct results [5]: another step is needed to verify the results. Image matching again was performed to assess the quality of the tracking results. Processing time for this step was much smaller than that of detection step since the size of the search space for the image matching was the same as that of the template. If the matching score is low (i.e., tracking failed), the camera switches its mode from tracking to detection in the following frame. Test results have shown that the processing speed is ten frames per second when all four cameras are in tracking mode.
3.2.4 Refinement

The refinement step is a significant process necessary to guarantee a reliable 3D positioning solution. Figure 3.9 illustrates the flow chart of the refinement process for a single camera. In this flow chart, the character D in a blue circle represents detection.
while T in a green circle represents tracking. The refinement step has two major functions. First, the refinement step helps cameras that have failed to find the target object to again find the target object by using the bundle adjustment result if it is available. This process is very important since it helps to keep or change the camera mode to tracking for faster processing. Second, refinement can remove incorrect matches.

<table>
<thead>
<tr>
<th>Index</th>
<th>Classification of a match</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>Detection</td>
</tr>
<tr>
<td>☑</td>
<td>Tracking</td>
</tr>
<tr>
<td>☑</td>
<td>Refined match that had a good matching score as well as large residual when # of good scores&gt;2 by current frame geometry</td>
</tr>
<tr>
<td>☑</td>
<td>Refined match that had a not-good matching score when # of good scores&gt;2 by current frame geometry</td>
</tr>
<tr>
<td>☑</td>
<td>Refined match that had a not-good matching score when # of good scores=2 by current frame geometry</td>
</tr>
<tr>
<td>☑</td>
<td>Refined match that had a not-good matching score when # of good scores&lt;2 by previous frame geometry</td>
</tr>
<tr>
<td>☑</td>
<td>Refined match that had a not-good matching score when # of good scores=2 by no geometry</td>
</tr>
</tbody>
</table>

Table 3.3. Classification of a match [73]

Assuming a camera fails to find the target object, if the bundle adjustment result is available using observations from other cameras, and if its estimated reference variance is good enough, then projecting the estimated 3D target object position to the camera that
has failed to find the object will provide a possible position of the target object in the camera. Then image matching with a smaller search space can provide the position of the object in the camera. If a bundle adjustment result is available, but its estimated reference variance is not good, or if a bundle adjustment result is not available, then the bundle adjustment result from the previous frame can be used, if it is available, to reduce the search space for image matching.

Although both detection and tracking have their own verification steps, in practice the results of both of these steps are not always correct. Assuming more than two cameras find the target object, then bundle adjustment will be available. If the estimated reference variance of the adjustment is good enough, then all observations are correct. Otherwise, it can be conjectured that there are possibly bad observations. In this case, bad observations can be found by performing secondary bundle adjustments with different sets of observations and monitoring their estimated reference variances. The observation set with the lowest estimate reference variance is chosen as the set of correct observations. It should be noted that not all observations with large residuals are incorrect. The observations with large residuals can be correct observations but be influenced by possible incorrect observations. In other words, an observation with a large residual also can be a correct observation.
3.3 Experiment

3.3.1 Detection, Tracking, and Refinement

A real-time system was designed and implemented as a proof of the concept for the algorithmic layout. This system was implemented in the C++ language with parallel processing and testing environment using Windows 7 running on a laptop computer having an Intel i7-3632QM CPU and 16 GB of RAM. The target area was a room measuring 8×5×2.5 meters with four GigE cameras installed at the each corner of the room. The implemented system estimated the 3D position of the target object at 6 to 10 fps processing speed. Note that the frame rate only for reading images from four cameras without any process is 16 fps. The fps values in Figure 3.11 to Figure 3.14 do not reflect the real-time values as the results were generated offline for the sake of providing visual results for this dissertation.

![Table of Camera IDs and Processing Modes](image)

Figure 3.10. Legend for the screenshots: camera ID (left) and information on a single camera screen (right) [73]
Figure 3.10 illustrates the legend for the screen shots shown in Figure 3.11 to Figure 3.14. The main screen consists of four images from each camera. Each image from each camera shows camera info, processing mode, bundle adjustment result, and the estimated 3D position of the target object. Table 3.3 describes the classification of a match shown in following screen shots.

Figure 3.11. Case 1 [73]: two cameras are in tracking mode and two cameras are in detection mode; two wrong matches are automatically corrected by refinement process.

Figure 3.11 shows the case when CAM0 and CAM2 are in tracking mode (blue rectangles) while CAM1 and CAM3 are in detection mode (red rectangles). CAM0
tracked the wrong object and CAM1 detected the wrong objects. The bundle adjustment was performed using matches with good matching scores (CAM1, CAM2, and CAM3). Since the square root of the estimated reference variance of the adjustment (±25.2 pixel) is higher than the predefined threshold (±5.0 pixel), the refinement process decided that the detection result for CAM1 was incorrect by monitoring estimated reference variances of different observation sets. The yellow rectangle in CAM0 and the green rectangle in CAM1 are refined results after the secondary bundle adjustment.

Figure 3.12. Case 2 [73]: all cameras are in tracking mode; however, only two correct observations are automatically selected to be used for bundle adjustment.
Figure 3.12 shows the case when all cameras are in tracking mode. Only the observations from CAM0 and CAM3 were used for the bundle adjustment. Incorrect observations were later refined by using the geometry of the previous frame because the square root of the estimated reference variance of the adjustment of the current frame (±6.6 pixel) was larger than the predefined threshold while that of the previous frame (±4.5 pixel, not shown in the figure) was smaller than the threshold.

Figure 3.13. Case 3 [73]: all cameras are in tracking mode; however, there is only one correct match, therefore, bundle adjustment is not available.
Similar to Figure 3.12 for Case 2, Figure 3.13 shows an example when all four cameras are in *tracking* mode (blue rectangles). However, only a match from CAM0 has a good matching score: the others do not. Therefore bundle adjustment at this frame was not available. In this case, the system used the bundle adjustment result from the previous frame that was available since it is assumed that the motion of the target object is small between consecutive frames. Salmon-colored rectangles in CAM1, CAM2, and CAM3 show refined matches by using the adjustment result of the previous frame.

Figure 3.14. Case 4 [73]: all cameras are in *tracking* mode; refinement process finds two correct matches and correct two wrong matches using bundle adjustment result.
Finally, Figure 3.14 shows an example of all four cameras in tracking mode with the tracked object in CAM0 and CAM2 have good matching scores. The observations in these two cameras were used for the bundle adjustment. Since the adjustment result showed a low square root of the estimated reference variance (±0.6 pixel), the adjustment result was used to refine the observations from the other two cameras. Cyan rectangles in CAM1 and CAM3 show the refined results.

The target object was moved randomly within the fields of view (FOVs) of the cameras within 1022 frames (150 sec; 6.8 fps). The BWB was successfully detected in 930 frames, which represents a 91% of success rate. Figure 3.15 illustrates the trajectory of
the BWB: change in the color of the trajectory represents time increments. Small sections with abrupt changes in the trajectory show where the system failed to detect BWB.

Figure 3.16. Mock-up for accuracy and precision test

To evaluate the accuracy and precision of the real-time 3D position estimation of the BWB, a mock-up (Figure 3.16) was created and placed at four different positions. The reason for using a mock-up was that the drill itself was too heavy to fix securely to the tripod. The 3D positions of the center of the BWB of the mock-up at each position were measured manually and the measured 3D coordinates were compared with the real-time estimations of the 3D coordinates from the system. Figure 3.17 shows a screen capture of the real-time estimation results of the 3D coordinates of the BWB.
Figure 3.18 illustrates differences between the measured and estimated 3D coordinates of the BWB at different positions. This experiment found that the maximum differences between measured and estimated 3D coordinates in all X, Y, and Z directions were ±6 mm, ±4 mm, and ±2 mm, respectively. The average differences in X, Y, and Z directions were ±4.75 mm, ±3.00 mm, and ±1.25 mm, respectively. Differences in the Z direction were relatively smaller (better) than for the other directions because of the linear shape of the BWB pattern.
Figure 3.18. Real-time BWB positioning accuracy (difference)

Figure 3.19 illustrates the standard deviations (1\(\sigma\)) of the adjustments. The standard deviations of 3D position estimation at all locations and in all \(X\), \(Y\), and \(Z\) directions were found to be less than \(\pm 5\ mm\).

Figure 3.19. Real-time BWB position precision (standard deviation)


3.3.2 Real-Time Bit Tip Positioning

The 3D position of the tip of the drill bit was estimated from the estimated BWB position, as shown in Figure 3.20. Since the length of the drill bit was known, a sphere could be drawn whose center was located at the center of the BWB and whose radius was the length of the drill bit. If the angle of the drill bit can be known in at least two different cameras, those 2D angles can be back-projected to find their intersection with the sphere in 3D.

Figure 3.20. Conceptual diagram of drill bit tip 3D positioning

A cone-shape target was installed and its 3D coordinates were measured using a Total Station to validate the accuracy of real-time estimation of the 3D coordinates of the drill bit tip. Figure 3.21 shows the installed target.
A video was recorded that showed the current position of the drill bit tip and positional differences between the installed target and the tip. This video was analyzed later because it was hard to see the estimated values in real-time. Figure 3.22 shows a frame of the recorded video. Note that all numbers displayed in the figure were estimated in real-time. In the left-hand side of the figure (white background), the $XYZ$ coordinates in black show the current position of the tip. The $XYZ$ coordinates in red show the differences between the measured target coordinates and the estimated bit tip coordinates. These differences were $2\ mm$, $−3\ mm$, and $−3\ mm$ in the $X$, $Y$, and $Z$ directions, respectively. Three rectangles at the lower right of the figure show views of the $XZ$, $YZ$, and $XY$ planes, respectively. In these views, a blue circle represents the installed target, a black circle represents the estimated BWB position, and a red line represents the drill bit. The big red crosshairs on the main screen represent the calculated position of the 3D drill bit tip that was projected to each camera.
Figure 3.23 illustrates accuracy evaluation results of the real-time drill bit tip 3D positioning experiment. The accuracy was calculated from positional differences between the measured 3D coordinates of the target and the estimated 3D coordinates of the drill bit tip when that tip is close to the target. Ten frames were chosen from the recorded video with which to analyze the accuracy. The maximum differences in the $X$, $Y$, and $Z$ directions were $\pm 7 \text{ mm}$, $\pm 8 \text{ mm}$, and $\pm 14 \text{ mm}$, respectively. The average differences of these ten frames in the $X$, $Y$, and $Z$ directions were $\pm 3.9 \text{ mm}$, $\pm 3.6 \text{ mm}$, and $\pm 9.3 \text{ mm}$, respectively. Difference in the $Z$ direction showed a relatively higher value than those for $X$ and $Y$ because there can be a small space between the target and the bit tip. Average
differences in the $X$ and $Y$ directions were less than $\pm 4\ mm$, which was an acceptable value since the thickness of the drill bit was $20\ mm$.

![Bit Tip Position Accuracy](image)

Figure 3.23. Real-time drill bit tip position accuracy

### 3.4 Conclusion

A multiple-camera system for object detection, tracking, and 3D positioning in real-time was designed and implemented such that it was able to run on a single laptop computer. The approach to implementing this system can be divided into the detection, tracking, and refinement stages. The main contribution of this approach is that geometric constraints can be used to refine both detection as well as tracking results while at the same time providing precise 3D positions of the target object. Experiments have shown
that the system can provide less than ±5 \textit{mm}, on average, real-time 3D positioning accuracy at speeds of 6 to 10 frames per second. The implementation and experiments conducted have demonstrated that this system can be used for real-time 3D positioning of objects.
4. MOTION FROM GEOMETRY

In this chapter, the ‘collinearity condition equations of motion’ and the ‘tracking from geometry’ methods developed in this research are introduced. The collinearity condition equations of motion represent the geometric relationship between 3D motion of a point in 3D object space and 2D motion of its image in 2D photo space. The collinearity equations of motion were derived from the collinearity equations. The proposed tracking from geometry method uses the collinearity equations of motion and the optical flow constraint equation to calculate 3D motion as well as 2D motions simultaneously from image sequences. Experiment results with simulated and real data show that both methods calculate 3D motion correctly.

4.1 Collinearity Condition of Motion

The collinearity condition of motion, as named in this dissertation research, represents the geometric relationship between the 3D motion of a point in 3D object space and the 2D motions of its image in 2D photo space. In other words, the collinearity condition of motion is the condition where a 3D point in object space after motion, its image on a photograph, and the perspective center of the photograph are all on a straight line, in addition to the collinearity condition (see Figure 4.1).
4.1.1 Collinearity Condition Equations of Motion

The collinearity condition equations of motion can be derived from the collinearity condition equations by subtracting two sets of equations or by taking derivatives of the equations with respect to time. The collinearity condition equations (4.1), which are the most well-known, the most important, and the most useful equations in photogrammetry, represent the geometric relationship between a point in 3D object space \((X_d, Y_d, Z_d)\) and its image in 2D photo space \((x_a, y_a)\).

\[
x_a = x_o - f \left[ \frac{m_{31}(X_d - X_L) + m_{32}(Y_d - Y_L) + m_{33}(Z_d - Z_L)}{m_{31}(X_d - X_L) + m_{32}(Y_d - Y_L) + m_{33}(Z_d - Z_L)} \right] = x_o - f \frac{r}{q} \\
y_a = y_o - f \left[ \frac{m_{31}(X_d - X_L) + m_{32}(Y_d - Y_L) + m_{33}(Z_d - Z_L)}{m_{31}(X_d - X_L) + m_{32}(Y_d - Y_L) + m_{33}(Z_d - Z_L)} \right] = y_o - f \frac{s}{q}
\] (4.1)
The variables $r$, $s$, and $q$ can be written as equation (4.2) from equation (4.1).

\[
\begin{align*}
    r &= m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L) \\
    s &= m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L) \\
    q &= m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)
\end{align*}
\] (4.2)

Since, $X_L$, $Y_L$, $Z_L$ (exposure station coordinates) and $m$’s (functions of the three rotation angles $\omega$, $\varphi$, and $\kappa$) in the above equations are constant for a stationary camera, the time derivatives of variables $r$, $s$, and $q$ are be rewritten as equation (4.3). Here, a dot above a variable denotes first order differentiation with respect to time [74].

\[
\begin{align*}
    \dot{r} &= m_{11}\dot{X}_A + m_{12}\dot{Y}_A + m_{13}\dot{Z}_A \\
    \dot{s} &= m_{21}\dot{X}_A + m_{22}\dot{Y}_A + m_{23}\dot{Z}_A \\
    \dot{q} &= m_{31}\dot{X}_A + m_{32}\dot{Y}_A + m_{33}\dot{Z}_A
\end{align*}
\] (4.3)

Two sets of collinearity equations can be written as equation (4.4). Then the motion equations can be derived by subtracting the two sets of collinearity equations.

\[
\begin{align*}
    x_a &= x_o - f \frac{r_A}{q_A} \\
    y_a &= y_o - f \frac{s_A}{q_A} \\
    x_b &= x_o - f \frac{r_B}{q_B} \\
    y_b &= y_o - f \frac{s_B}{q_B}
\end{align*}
\] (4.4)

79
Subtracting point \( A \) from point \( B \) in the photo space gives equation (4.5).

\[
\begin{align*}
    x_b - x_a &= -f \left( \frac{r_b - r_a}{q_b} \right) = -f \left( \frac{r_b q_A - r_A q_b}{q_A q_b} \right) \\
    y_b - y_a &= -f \left( \frac{s_b - s_a}{q_b} \right) = -f \left( \frac{s_b q_A - s_A q_b}{q_A q_b} \right)
\end{align*}
\]  

(4.5)

The variables \( r_b \), \( s_b \), and \( q_b \) are written as followings.

\[
\begin{align*}
    r_b &= r_A + \dot{r}_A dt \\
    s_b &= s_A + \dot{s}_A dt \\
    q_b &= q_A + \dot{q}_A dt
\end{align*}
\]  

(4.6)

Substituting the above equations into equations (4.5) gives equation (4.7).

\[
\begin{align*}
    x_b - x_a &= -f \left( \frac{r_A + \dot{r}_A dt}{q_A} \right) = -f \left( \frac{\dot{r}_A q_A - r_A \dot{q}_A dt}{q_A} \right) \\
    y_b - y_a &= -f \left( \frac{s_A + \dot{s}_A dt}{q_A} \right) = -f \left( \frac{\dot{s}_A q_A - s_A \dot{q}_A dt}{q_A} \right)
\end{align*}
\]  

(4.7)

Then motions in the photo space are written as equations (4.8).

\[
\begin{align*}
    \dot{x}_a &= \frac{x_b - x_a}{dt} = -f \left( \frac{\dot{r}_A q_A - r_A \dot{q}_A dt}{q_A} \right) \\
    \dot{y}_a &= \frac{y_b - y_a}{dt} = -f \left( \frac{\dot{s}_A q_A - s_A \dot{q}_A dt}{q_A} \right)
\end{align*}
\]  

(4.8)
Assuming that $dt$ is an infinitesimally small unit of time, then equations (4.8) are written as equations (4.9).

\[
\dot{x}_a = -f \frac{\dot{r}_a q_a - r_a \dot{q}_a}{q_a(q_a + \dot{q}_a)} \\
\dot{y}_a = -f \frac{\dot{s}_a q_a - s_a \dot{q}_a}{q_a(q_a + \dot{q}_a)}
\]

(4.9)

In this dissertation, equations (4.9) are called the ‘collinearity condition equations of motion.’ They represent the geometric relationship between the 3D motion of a point in 3D object space and the 2D motions of its image in 2D image space. Prerequisite constants for the equations are the exterior orientation parameters of a stationary camera ($X_L, Y_L, Z_L, \omega, \varphi, \text{ and } \kappa$), the 3D coordinates of a point at time $t_0$ ($X, Y, \text{ and } Z$), and the focal length ($f$).

**4.1.2 Linearization of the Collinearity Equations of Motion**

The nonlinear collinearity equations of motion are linearized by using Taylor’s theorem.

\[
F = -f \frac{\ddot{q} - \dot{q}^2}{q(q + \dot{q})} = \dot{x} \\
G = -f \frac{\ddot{s} - \dot{s}^2}{q(q + \dot{q})} = \dot{y}
\]

(4.10)
The linearized form of the collinearity equations of motion is written as equation (4.11) by taking partial derivatives of equation (4.10) with respect to the unknowns.

\[
F_0 + \left( \frac{\partial F}{\partial X} \right)_0 d\dot{X} + \left( \frac{\partial F}{\partial Y} \right)_0 d\dot{Y} + \left( \frac{\partial F}{\partial Z} \right)_0 d\dot{Z} = \dot{x}
\]

\[
G_0 + \left( \frac{\partial G}{\partial X} \right)_0 d\dot{X} + \left( \frac{\partial G}{\partial Y} \right)_0 d\dot{Y} + \left( \frac{\partial G}{\partial Z} \right)_0 d\dot{Z} = \dot{y}
\]

Equations (4.11) are simplified as equations (4.12).

\[
M = h_{11} d\dot{X} + h_{12} d\dot{Y} + h_{13} d\dot{Z} + e_x
\]

\[
N = h_{21} d\dot{X} + h_{22} d\dot{Y} + h_{23} d\dot{Z} + e_y
\]

where

\[
h_{11} = f \frac{m_{11} q(r + \dot{r}) - m_{12} q(q + \dot{q})}{q(q + \dot{q})^2} \quad h_{12} = f \frac{m_{12} q(r + \dot{r}) - m_{11} q(q + \dot{q})}{q(q + \dot{q})^2} \]

\[
h_{13} = f \frac{m_{13} q(r + \dot{r}) - m_{11} q(q + \dot{q})}{q(q + \dot{q})^2} \quad M = \dot{x} + f \frac{\dot{r}q - r\dot{q}}{q(q + \dot{q})}
\]

\[
h_{21} = f \frac{m_{21} q(s + \dot{s}) - m_{22} q(q + \dot{q})}{q(q + \dot{q})^2} \quad h_{22} = f \frac{m_{22} q(s + \dot{s}) - m_{21} q(q + \dot{q})}{q(q + \dot{q})^2} \]

\[
h_{23} = f \frac{m_{23} q(s + \dot{s}) - m_{22} q(q + \dot{q})}{q(q + \dot{q})^2} \quad N = \dot{y} + f \frac{\dot{s}q - s\dot{q}}{q(q + \dot{q})}
\]
4.1.3 Space Intersection of Motion

Space intersection is the photogrammetric procedure that estimates coordinates of a point in 3D object space from photo coordinates of observations in a stereo pair. Likewise, the space intersection of motion estimates the motion of a point in the 3D object space from motion observations in 2D photo space in a stereo pair. This is represented as equation (4.13).

\[
\begin{bmatrix}
(M)_{C_0} \\
(N)_{C_0} \\
(M)_{C_1} \\
(N)_{C_1}
\end{bmatrix}
= 
\begin{bmatrix}
(h_{11})_{C_0} & (h_{12})_{C_0} & (h_{13})_{C_0} \\
(h_{21})_{C_0} & (h_{22})_{C_0} & (h_{23})_{C_0} \\
(h_{11})_{C_1} & (h_{12})_{C_1} & (h_{13})_{C_1} \\
(h_{21})_{C_1} & (h_{22})_{C_1} & (h_{23})_{C_1}
\end{bmatrix}
\begin{bmatrix}
\frac{dX_A}{d} \\
\frac{dY_A}{d} \\
\frac{dZ_A}{d}
\end{bmatrix}
+ e
\] (4.13)

In equation (4.13), subscripts \(C_0\) and \(C_1\) represent parameters related to camera 0 and camera 1, respectively. The space intersection of motion method can calculate the 3D motion iteratively from 2D motions.

4.2 Tracking from Geometry

The ‘tracking from geometry’ method, as named in this dissertation research, simultaneously finds 2D motions in image planes and 3D motion in 3D object space using multiple cameras by combining the collinearity equations of motion and the optical flow constraint equation. The optical flow constraint equation can estimate 2D motion in image sequences by using, for example, the Lucas-Kanade method. The collinearity
condition equations of motion define the relationship between 2D motion and 3D motion. Therefore, 3D motion, as well as 2D motions, can be estimated directly from multiple image sequences.

4.2.1 Optical Flow Constraint Equation

The optical flow constraint equation is repeated here for convenience. The optical flow constraint equation is written as equation (4.14).

\[ I_x u + I_y v = -I_t \] (4.14)

Assuming that motion of neighboring pixels is same, then the optical flow constraint equations for a small area that consists of neighboring \( n \) number of pixels are written as equation (4.15).

\[
\begin{bmatrix}
I_{x1} & I_{y1} \\
I_{x2} & I_{y2} \\
\vdots & \vdots \\
I_{xn} & I_{yn}
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= 
\begin{bmatrix}
-I_{t1} \\
-I_{t2} \\
\vdots \\
-I_{tn}
\end{bmatrix}
\] (4.15)
4.2.2 Photo and Image Coordinate Systems

Since the pixel is the unit of motion in an image coordinate system in the optical flow constraint equations, it is necessary to convert the image coordinate units [pixel] to photo coordinate units [mm] in order to combine the optical flow constraint equations with the collinearity condition equations of motion. Image coordinates can be converted to measured photo coordinates, which means uncalibrated photo coordinates, by using equations (4.16).

\[ x' = \left( c - \frac{n\text{Cols}}{2} \right) \times PS \]
\[ y' = \left( \frac{n\text{Rows}}{2} - r \right) \times PS \]  

Equations (4.17) represent the inverses of equations (4.16).

\[ c = \frac{x'}{PS} + \frac{n\text{Cols}}{2} \]
\[ r = \frac{n\text{Rows}}{2} - \frac{y'}{PS} \]  

Since \( PS \), \( n\text{Cols} \), and \( n\text{Rows} \) are constants in equations (4.17), taking derivatives of equations (4.17) with respect to time gives equations (4.18), where a dot above a variable denotes first order differentiation with respect to time.

\[ \dot{c} = \frac{\dot{x}'}{PS} \]
\[ \dot{r} = -\frac{\dot{y}'}{PS} \]  

85
Equations (4.18) show the relationship between motions of a point in an image coordinate system and in a photo coordinate system.

\[ \dot{\varepsilon} = u \quad \dot{r} = v \]  

(4.19)

The image motion terms \((u, v)\) in both the optical flow constraint equation and the Lucas-Kanade method correspond to \(\dot{\varepsilon}\) and \(\dot{r}\) in equations (4.19).

### 4.2.3 The Collinearity Equations of Motion in Image Space

The collinearity condition equations of motion are written as equation (4.20).

\[
\begin{bmatrix}
M \\
N
\end{bmatrix} = 
\begin{bmatrix}
\dot{x} + f \frac{\dot{q} - r \dot{q}}{q(q + \dot{q})} \\
\dot{y} + f \frac{s q - s \dot{q}}{q(q + \dot{q})}
\end{bmatrix} = 
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23}
\end{bmatrix} 
\begin{bmatrix}
d\hat{X}_A \\
d\hat{Y}_A \\
d\hat{Z}_A
\end{bmatrix} + e
\]  

(4.20)

Equation (4.20) represents the relationship between motions in the 3D object coordinate system and in the 2D photo coordinate system. Assuming an image does not have any distortion, the relationship between 3D motion in 3D object space and 2D motion in image space is then written as equation (4.21).
4.2.4 3D Motion from Image Sequence

The optical flow constraint equation for a small area that consists of \( n \) number of neighboring pixels is written as equation (4.22).

\[
\begin{bmatrix}
I_{x_{i1}} & I_{y_{i1}} \\
I_{x_{i2}} & I_{y_{i2}} \\
\vdots & \vdots \\
I_{x_{in}} & I_{y_{in}}
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} =
\begin{bmatrix}
-I_{i1} \\
-I_{i2} \\
\vdots \\
-I_{in}
\end{bmatrix}
\] (4.22)

Since \((u, v)\) in the Lucas-Kanade method is practically \((du, dv)\), substituting equation (4.21) into the optical flow constraint equation provides equation (4.23).

\[
\frac{1}{PS}
\begin{bmatrix}
I_{x_{i1}} & I_{y_{i1}} \\
I_{x_{i2}} & I_{y_{i2}} \\
\vdots & \vdots \\
I_{x_{in}} & I_{y_{in}}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & -1 \\
\vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23} \\
\vdots & \vdots & \vdots
\end{bmatrix}
\begin{bmatrix}
dX_A \\
dY_A \\
dZ_A
\end{bmatrix}
+ e' =
\begin{bmatrix}
-I_{i1} \\
-I_{i2} \\
\vdots \\
-I_{in}
\end{bmatrix}
\] (4.23)
Equation (4.23) also is written as equation (4.24).

\[
\frac{1}{PS} \begin{bmatrix}
   h_{11} I_{x_1} - h_{21} I_{y_1} & h_{12} I_{x_1} - h_{22} I_{y_1} & h_{13} I_{x_1} - h_{23} I_{y_1} \\
   h_{11} I_{x_2} - h_{21} I_{y_2} & h_{12} I_{x_2} - h_{22} I_{y_2} & h_{13} I_{x_2} - h_{23} I_{y_2} \\
   \vdots & \vdots & \vdots \\
   h_{11} I_{x_n} - h_{21} I_{y_n} & h_{12} I_{x_n} - h_{22} I_{y_n} & h_{13} I_{x_n} - h_{23} I_{y_n}
\end{bmatrix}
\begin{bmatrix}
   d\hat{X}_d \\
   d\hat{Y}_d \\
   d\hat{Z}_d
\end{bmatrix} = \begin{bmatrix}
   -I_n \\
   -I_{t_2} \\
   \vdots \\
   -I_{t_n}
\end{bmatrix} + e'
\]  

(4.24)

The unknown value to be calculated is the 3D motion. LESS is used to calculate the unknown 3D motion. The design matrix \((A)\) and the observation matrix \((y)\) that are used to solve the LESS are shown in equation (4.25). The size of the normal matrix \((N)\) is \(3\times3\).

The normal matrix can be directly calculated for faster processing.

\[
A = \frac{1}{PS} \begin{bmatrix}
   h_{11} I_{x_1} - h_{21} I_{y_1} & h_{12} I_{x_1} - h_{22} I_{y_1} & h_{13} I_{x_1} - h_{23} I_{y_1} \\
   h_{11} I_{x_2} - h_{21} I_{y_2} & h_{12} I_{x_2} - h_{22} I_{y_2} & h_{13} I_{x_2} - h_{23} I_{y_2} \\
   \vdots & \vdots & \vdots \\
   h_{11} I_{x_n} - h_{21} I_{y_n} & h_{12} I_{x_n} - h_{22} I_{y_n} & h_{13} I_{x_n} - h_{23} I_{y_n}
\end{bmatrix} \begin{bmatrix}
   -I_n \\
   -I_{t_2} \\
   \vdots \\
   -I_{t_n}
\end{bmatrix}
\]  

(4.25)

The solution for the LESS is written as equation (4.26).

\[
\begin{bmatrix}
   d\hat{X}_d \\
   d\hat{Y}_d \\
   d\hat{Z}_d
\end{bmatrix} = (A^T A)^{-1} A^T y
\]  

(4.26)
4.2.5 Multiple Camera Case

To calculate 3D motion directly from sets of 2D motions, multiple cameras are necessary. Equation (4.24) is written as equation (4.27) for the multiple-camera case.

\[
\begin{align*}
\begin{pmatrix}
(I_{x_0})_{c_0} & (I_{y_0})_{c_0} & 0 & 0 \\
(I_{x_1})_{c_0} & (I_{y_1})_{c_0} & 0 & 0 \\
0 & 0 & (I_{x_0})_{c_1} & (I_{y_0})_{c_1} \\
0 & 0 & (I_{x_1})_{c_1} & (I_{y_1})_{c_1}
\end{pmatrix}
& \begin{pmatrix}
(h_{x_0})_{c_0} & (h_{y_0})_{c_0} & (h_{y_0})_{c_0} \\
(h_{x_1})_{c_0} & (h_{y_1})_{c_0} & (h_{y_1})_{c_0} \\
-(h_{x_0})_{c_1} & -(h_{y_0})_{c_1} & -(h_{y_0})_{c_1} \\
-(h_{x_1})_{c_1} & -(h_{y_1})_{c_1} & -(h_{y_1})_{c_1}
\end{pmatrix}
\begin{pmatrix}
\frac{dx}{dt} \\
\frac{d\hat{Y}}{dt} \\
\frac{dZ}{dt}
\end{pmatrix}
\end{align*}
\]

\[
= \begin{pmatrix}
-I_{x_0, c_0} \\
-I_{y_0, c_0} \\
-I_{x_1, c_1} \\
-I_{y_1, c_1}
\end{pmatrix} + e' \quad (4.27)
\]

In equation (4.27), subscripts \(C_0\) and \(C_1\) represent parameters related to camera 0 and camera 1, respectively. Regardless of the number of cameras used, the size of normal matrix \((N)\) is 3\(\times\)3.

4.3 Experiment

In this section, the experimental results for the proposed collinearity condition equations of motion and for the proposed tracking from geometry method using simulated as well as real data sets will be shown. The test environment including camera setup is discussed in Section 4.3.1. The experiment for the collinearity condition equations of motion is discussed in Section 4.3.2. As outlined in this chapter, the 3D positions of a simulated moving object were calculated using a stereo camera and the proposed collinearity condition equations of motion. In Section 4.3.3, direct calculation of 2D and 3D motion...
of a moving object in a real image sequence using the proposed tracking from geometry method is discussed.

4.3.1 Interior and Exterior Orientation of Cameras

Figure 4.2. Stereo camera system built for experiments

A stereo camera system (Figure 4.2) was built to test the proposed collinearity condition equations of motion and the tracking from geometry methods. Two Gigabit Ethernet (GigE) cameras (CAM4 and CAM5) were tightly affixed to a square beam to provide precise stereo geometry. The length of the baseline (distance between camera centers) was 0.7091 m. The interior orientation parameters of each camera were estimated using the photogrammetric method described in Section 2.2.4. They are listed in Table 4.1. The size of a pixel in these cameras was 0.0035×0.0035 mm.
Figure 4.3 shows one of acquired images for camera calibration (left) and its distortion corrected image (right). The left image clearly shows symmetric radial lens distortion. The right image was generated by resampling each pixel of the original image with the estimated lens distortion parameters using the left part of equation (2.18).

![Figure 4.3. Image with lens distortion (left) and distortion corrected image (right) using the estimated lens distortion parameters](image)

Table 4.1. The estimated interior orientation parameters for CAM4 and CAM5

<table>
<thead>
<tr>
<th>IOP</th>
<th>CAM4</th>
<th>CAM5</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>4.5495 mm</td>
<td>4.6224 mm</td>
</tr>
<tr>
<td>x₀</td>
<td>3.5011e-2 mm</td>
<td>8.9037e-3 mm</td>
</tr>
<tr>
<td>y₀</td>
<td>4.4867e-2 mm</td>
<td>2.1558e-2 mm</td>
</tr>
<tr>
<td>k₁</td>
<td>1.4883e-2</td>
<td>1.5694e-2</td>
</tr>
<tr>
<td>k₂</td>
<td>3.6041e-4</td>
<td>2.0851e-4</td>
</tr>
<tr>
<td>k₃</td>
<td>-1.0829e-6</td>
<td>9.9868e-6</td>
</tr>
<tr>
<td>p₁</td>
<td>-1.4092e-4</td>
<td>-2.4110e-4</td>
</tr>
<tr>
<td>p₂</td>
<td>3.1624e-6</td>
<td>-9.7042e-5</td>
</tr>
</tbody>
</table>

Figure 4.3 shows one of acquired images for camera calibration (left) and its distortion corrected image (right). The left image clearly shows symmetric radial lens distortion. The right image was generated by resampling each pixel of the original image with the estimated lens distortion parameters using the left part of equation (2.18).
The exterior orientation parameters of this stereo camera system were estimated using the photogrammetric bundle adjustment method described in Section 2.2.9. They are listed in Table 4.2.

<table>
<thead>
<tr>
<th>EOP</th>
<th>CAM4</th>
<th>CAM5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$ [rad]</td>
<td>$-1.54878322 \pm 0.00330527$</td>
<td>$-1.43827898 \pm 0.01054644$</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>$1.83223519 \pm 0.00052720$</td>
<td>$1.65302810 \pm 0.00051742$</td>
</tr>
<tr>
<td>$\kappa$ [rad]</td>
<td>$-3.14326051 \pm 0.00338112$</td>
<td>$-3.24801353 \pm 0.01058512$</td>
</tr>
<tr>
<td>$X_L$ [m]</td>
<td>$148.93574 \pm 0.00074$</td>
<td>$149.08220 \pm 0.00051$</td>
</tr>
<tr>
<td>$Y_L$ [m]</td>
<td>$245.67864 \pm 0.00174$</td>
<td>$246.37230 \pm 0.00188$</td>
</tr>
<tr>
<td>$Z_L$ [m]</td>
<td>$351.23461 \pm 0.00273$</td>
<td>$351.24961 \pm 0.00278$</td>
</tr>
</tbody>
</table>

Table 4.2. The estimated exterior orientation parameters for CAM4 and CAM5

4.3.2 Collinearity Condition Equations of Motion

The collinearity condition equations of motion proposed in this dissertation provide the relationship between 2D motion and 3D motion. From these equations, another form of equations can be derived (which is the space intersection of motion described in Section 4.1.3) to calculate 3D motion iteratively from a stereo pair of 2D motions. Assuming that the 3D positions of a moving object need to be estimated with a stereo camera system at every frame and those 2D motions are observed correctly at every frame, then the 3D positions of the moving object at every frame can be calculated with 2D motion observations and the initial 3D position of the object.
<table>
<thead>
<tr>
<th>Number of Frames</th>
<th>553</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Object Start Point [m]</td>
<td>(140.0, 249.0, 352.0)</td>
</tr>
<tr>
<td>Target Object End Point [m]</td>
<td>(144.0, 245.5, 350.5)</td>
</tr>
<tr>
<td>Target Object Travel Length [m]</td>
<td>5.523</td>
</tr>
</tbody>
</table>

Table 4.3. Simulation data for space intersection of motion

Table 4.3 lists the conditions for a simulation experiment. An object moves from (140.0 m, 249.0 m, 352.0 m) to (144.0 m, 245.5 m, 350.5 m) and the stereo camera system takes 553 stereo pairs of images of a moving object that travels 5.523 m. It is assumed that the motions in the image planes are correctly observed in every frame. Figure 4.4 illustrates the positions of the cameras (blue) and the trajectory of the object (red).

Figure 4.4. The position of the stereo camera system (blue) and trajectory of a moving object (red)
Figure 4.5 shows 3D motion error with respect to frame number. The proposed method needs only the 3D position of an object at the beginning. It then calculates 3D motion continuously at each frame from the previously calculated 3D position. The test results showed that 3D motion error was negligible.

![Error in 3D Motion with respect to Frame Number](image)

Figure 4.5. Calculated 3D motion error with respect to frame number

Figure 4.6 illustrates 3D position error with respect to frame number. The 3D position error is accumulated since the proposed method calculates 3D position from previously calculated 3D position. Even though these position errors are negligible, it needs to be corrected using the real 3D position that is used for the initial value in the proposed method if necessary. These simulation experiments showed that the proposed method can calculate 3D motion of a moving object with an acceptable level of positioning error. In a real case scenario, this amount of error at each frame is negligible because of limited subpixel accuracy in 2D motion observations.
4.3.3 Tracking from Geometry

The proposed tracking from geometry method can estimate 2D motions and 3D motion simultaneously by using the collinearity condition equations of motion and the optical flow constraint equation. This method only needs the initial 2D position in each image from the stereo cameras and the initial 3D position of an object to simultaneously calculate 2D and 3D motion.
The stereo camera system used here is described in Section 4.3.1. The target object for this experiment was a marker taped to a book (Figure 4.7). The marker was used for triangulation of the Lucas-Kanade only results (2D tracking). Two-dimensional motion of the target in the images and 3D motion were calculated simultaneously using equation (4.27). The target object moved 0.1956 m in 90 frames. Figure 4.9 shows the position of the target object (red crosshair) at the end of tracking. Final positions of the target object in image space were same in both tracking from geometry (3D tracking) and Lucas-Kanade (2D tracking).

<table>
<thead>
<tr>
<th></th>
<th>X [m]</th>
<th>Y [m]</th>
<th>Z [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Tracking:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tracking from</td>
<td>146.726187</td>
<td>246.223044</td>
<td>351.503914</td>
</tr>
<tr>
<td>Geometry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2D Tracking:</td>
<td>146.726186</td>
<td>246.223044</td>
<td>351.503907</td>
</tr>
<tr>
<td>Lucas-Kanade</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>then Triangulation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difference</td>
<td>0.000001</td>
<td>0.000000</td>
<td>0.000007</td>
</tr>
</tbody>
</table>

Table 4.4. Comparison of 3D positions estimated by 2D tracking (Lucas-Kanade then triangulation) method and 3D tracking (tracking from geometry) method at the last frame.
Table 4.4 lists the results of a comparison between the calculated 3D positions at the last frame (frame 189) and Figure 4.8 illustrates the difference in the calculated 3D position between the 2D (Lucas-Kanade then triangulation) and 3D tracking (tracking from geometry) methods with respect to frame number.

![Difference in 3D Position between 2D and 3D Tracking Result](image)

Figure 4.8. Differences in calculated 3D position results between 2D tracking (Lucas-Kanade then triangulation) and 3D tracking (tracking from geometry) methods with respect to frame number.

The tracking from geometry method calculates 3D motion, therefore also the 3D position, simultaneously with the 2D motions. The 2D tracking method tracks the object using the Lucas-Kanade method, which gives 2D motions and then triangulates to estimate the 3D position using the photogrammetric space intersection method. The differences in position between the two methods were 0.001 mm, 0.000 mm, and 0.007 mm for the X, Y, and Z directions, respectively. These differences are negligible.
4.4 Conclusion

Both the ‘collinearity condition equations of motion’ and the ‘tracking from geometry’ methods were developed in this research. The collinearity equations of motion represent the geometric relationship between motions in 2D and 3D. The tracking from geometry method combines the collinearity equations of motion with the optical flow constraint equation to simultaneously estimate 2D motions in image sequences as well as directly
estimating 3D motion. The developed ‘tracking from geometry’ method together with the ‘collinearity equations of motion’ has the following advantage. If there is a need to estimate the precise 3D position of a moving object, the method does not need the additional triangulation step to estimate the 3D position of the object. The proposed method calculates 2D motions in images and 3D motion in object space simultaneously while other tracking methods need triangulation to estimate 3D position.
5. SUMMARY AND FUTURE WORK

5.1 Summary

This dissertation research had two primary goals.

I. Implementing a real-time object detection, tracking, and 3D positioning system using multiple cameras; and

II. Investigating the relationship between 2D motions and 3D motion, or ‘collinearity condition of motion’, which includes

i. Demonstrating direct calculation of 3D motion as well as 2D motions from image sequences using the ‘tracking from geometry’ method.

Firstly, a real-time multiple-camera (four) system object detection, tracking, and 3D positioning system which is running on a single laptop computer was designed and implemented. Also, positioning accuracy was evaluated using pre-measured 3D target positions for evaluation purposes. The system successfully showed reliable object detection, tracking, and 3D positioning results at update rates of 6 Hz to 10 Hz. All four cameras were precisely calibrated using a separately developed camera calibration
program. All four cameras were installed at the each corner of a room measuring $8 \times 5 \times 2.5$ meters and the exterior orientation parameters of these cameras were estimated using the bundle adjustment method with pre-installed control points. All cameras were connected to a processing computer via a network switch through Ethernet cables. The target object here was a drill which is used in a construction site. A feature to be tracked (BWB: black-white-black stripes wound around the chuck of the drill) was designed and rotation and scale invariant templates were generated for the BWB. Detected BWBs were evaluated by epipolar geometry between cameras since a high matching score does not always guarantee the correct matching result. Once the detected BWB satisfied epipolar geometry, the BWB was assumed to be the correct match, and then the camera changed its mode from detection to tracking for more rapid processing. Detection and tracking results were again evaluated by statistics such as estimated reference variance and residuals, which were estimated from bundle adjustment at each frame. This process is called refinement. From the 3D BWB position estimated by bundle adjustment, the 3D position of the drill bit tip was calculated in real-time. Experiment results showed an average real-time 3D positioning accuracy of less than $\pm 5mm$ at an update rate of between 6 Hz (when all cameras were in detection mode) and 10 Hz (when all cameras were in tracking mode). This implementation and experiment demonstrated that the developed system can be used for real-time 3D positioning of objects.
Secondly, collinearity condition equations of motion that represent the geometric relationship between 2D motion and 3D motion were derived. The collinearity equations of motion were derived from a pair of collinearity condition equations. Simulation experiments showed that the proposed method correctly calculates 3D motion from a pair of 2D motions. The tracking from geometry method combines the collinearity condition equations of motion with the optical flow constraint equation to estimate 2D motions and 3D motion simultaneously. The motion term in image space \((u, v)\) of the optical flow constraint equation was substituted for the motion term in photo space \((\dot{x}, \dot{y})\) to combine the optical flow equation with the collinearity equations of motion. Then the motion in image space estimated by the optical flow constraint can be used for the collinearity equations of motion. To calculate 3D motion using the tracking from geometry method, at least two cameras are needed. A stereo camera system was built to test the proposed method. The unknown values as well as the adjusted values in the equations for 2D tracking methods are 2D motions in image space. The unknown values as well as adjusted values in the tracking from geometry (3D tracking) method are motion \((\dot{X}, \dot{Y}, \dot{Z})\) in 3D space. The 2D motions in 2D image space also can be calculated by solving the collinearity equations of motion with the calculated 3D motion. The experiment with real image sequences showed that the proposed tracking from geometry method provides accurate 3D position results. The calculated 3D position was compared with the space intersection results of 2D tracking (Lucas-Kanade then triangulation) method. The differences between the two methods were less than \(\pm 0.01\ mm\) in all \(X, Y,\) and \(Z\).
directions. The advantage of the tracking from geometry method is that this method calculates 2D motions and 3D motion simultaneously while other tracking methods need a separate triangulation step to estimate 3D position.

In summary, this dissertation research makes the following contributions to the field:

- **Real-time multiple camera object detection, tracking, and 3D positioning system**
  - A real-time multiple camera object detection, tracking, and 3D positioning system was developed and tested that provides real-time 3D positioning accuracy of less than ±5 mm in average at update rates of 6 Hz to 10 Hz in a room measuring 8×5×2.5 meters.
  - A statistic refinement process was developed and tested that refines detection and tracking results (which do not always provide the correct solution) to maintain the reliability of 3D positioning using statistics from bundle adjustment.

- **Motion from Geometry**
  - The collinearity condition equations of motion were derived that define the geometric relationship between 2D and 3D motions.
  - The tracking from geometry method was developed that enables finding 3D motion and 2D motions simultaneously from a set of image sequences directly without triangulation.
5.2 Future Work

Although this real-time, multiple-camera object 3D positioning system shows reliable positioning results, the system has limitations. First, a special feature and its scale and rotation invariant templates had to be designed to be recognized in the cameras with different views. For general purposes, extensive study on recognition algorithms for arbitrary features at any distance and in any direction will be important. Secondly, tracking and detection are vulnerable to different illumination conditions. This is not only a problem of the proposed system, but also one of all vision-related algorithms. Lastly, the system provides a six-to-ten frames per second processing speed, which is insufficient for real-time processes if more cameras are added. The system needs to be calibrated more efficiently.

In this dissertation research, the collinearity condition equations of motion did not consider lens distortion. It was assumed that the lens had no distortion and provided distortion corrected image sequences for the tests. Tracking from geometry also has limitations in that the calculated 3D motion strongly depends on the accuracy of tracking since the geometry between the cameras is used only for establishing the relationship between 2D and 3D motion. Therefore, if the tracking failed, the 3D motion calculation also has failed. The addition of a geometric constraint that can guide the tracking process provides a future direction of investigation.
REFERENCES


Processing, 2006.


[37] S. Tariq and F. Dellaert, "A Multi-Camera 6-DOF Pose Tracker," in *IEEE and ACM*


2002.


[60] Schenk, "Introduction to Photogrammetry," CEEG, The Ohio State University, Columbus, 2005.


