UNIVERSITY OF CALIFORNIA, SAN DIEGO

Precise Estimation of the Geoposition and Orientation of Ground-level Video Cameras from Multiple Sensors

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, & Control)

by

Benjamin L. Ochoa

Committee in charge:

Professor Serge Belongie, Chair
Professor Nuno Vasconcelos, Co-Chair
Professor Pamela Cosman
Professor David Kriegman
Professor Truong Nguyen

2007

The dissertation of Benjamin L. Ochoa is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2007

To Tricia

TABLE OF CONTENTS

Signature Page . . . iii

Dedication . . . iv

Table of Contents . . . v

List of Figures . . . vii

List of Tables . . . viii

Acknowledgements . . . ix

Vita . . . x

Abstract of the Dissertation . . . xi

1 Overview . . . 1
- 1.1 Introduction . . . 1
- 1.2 Sensors . . . 5
  - 1.2.1 Video camera . . . 6
  - 1.2.2 GPS receiver . . . 7
  - 1.2.3 3-axis orientation sensor . . . 10
- 1.3 Single Camera Estimation . . . 16
  - 1.3.1 Sequential estimation of geoposition and orientation . . . 16
- 1.4 Multiple Camera Estimation . . . 21
  - 1.4.1 Feature detection and matching . . . 22
  - 1.4.2 Joint estimation of geoposition and orientation . . . 27
- 1.5 Experimental results . . . 28
- 1.6 Conclusions . . . 30

2 Sensors . . . 33
- 2.1 Video camera . . . 33
  - 2.1.1 Camera model . . . 34
  - 2.1.2 Camera calibration . . . 38
- 2.2 GPS receiver . . . 39
  - 2.2.1 Geodetic coordinate transformation . . . 40
  - 2.2.2 GPS positioning uncertainty . . . 43
- 2.3 3-axis orientation sensor . . . 44
  - 2.3.1 Rotation to camera coordinate frame and its uncertainty . . . 46

3 Single Camera Estimation . . . 51
- 3.1 Sequential estimation of geoposition and orientation . . . 52
  - 3.1.1 Motion estimation from video . . . 52
  - 3.1.2 Kalman filter . . . 55

4 Multiple Camera Estimation 61
4.1 Feature detection and matching 61
4.1.1 Covariance propagation for guided matching 62
4.1.2 Feature matching 71
4.2 Joint estimation of geoposition and orientation 75

A Appendix 78
A.1 Partial derivatives of matrix operations 78

Bibliography 82

LIST OF FIGURES

Figure 1.1: Sensors in the data acquisition system 5
Figure 1.2: Relationship between coordinates in the camera coordinate frame, image coordinates, and normalized coordinates 6
Figure 1.3: Relationship between geodetic and geocentric coordinates 9
Figure 1.4: Pitch-roll-yaw unrotated local Cartesian coordinate frame 12
Figure 1.5: Relationship between 3D coordinate frames 14
Figure 1.6: Feature detection and tracking 18
Figure 1.7: Features for matching 23
Figure 1.8: Point-to-line mapping under a fundamental matrix 26
Figure 1.9: Geoposition estimates and their uncertainty 30
Figure 1.10: Distance between geoposition measurement and estimate 31

Figure 2.1: Images of the camera calibration target 39

Figure 4.1: Point-to-point mapping under a planar homography 70
Figure 4.2: Mosaic construction from video 72

LIST OF TABLES

Table 1.1: Standard GPS error model 11

Table 2.1: Estimated internal camera parameters and their uncertainty 40
Table 2.2: Common reference ellipsoids and their parameters 42
Table 2.3: Tilt compensated compass-magnetometer sensor error 46

ACKNOWLEDGEMENTS

My time as a student at UCSD has finally concluded. Over my years of study, many people that have taught, helped, and encouraged me. To all of them, I am grateful.

Foremost, I thank my advisor, Serge Belongie. I have the utmost respect for Serge as a teacher, researcher, and individual. It has been a privilege to work with him and I look forward to our continued collaboration and friendship. I also thank Ramesh Jain for introducing me to the field of computer vision and its possibilities, and for encouraging me to attend graduate school.

Spending time in the Computer Vision Laboratory with Sameer Agarwal, Kristin Branson, Manmohan Chandraker, Piotr Dollar, Vincent Rabaud, Satya Mallick, Andrew Rabinovich, and Josh Wills has been a pleasure. Without a doubt, my work has been improved due to our many conversations. Thanks to you all for providing such a stimulating environment. Thanks also to John Dolloff, Jim Olson, and Alan Sussman at BAE Systems for helpful discussions. Also, Pamela Cosman and David Kriegman provided excellent comments on earlier drafts of this dissertation.

Finally, I dedicate this dissertation to my wife, Tricia. Without her love, understanding, and support, I would not have been able to attend graduate school. I also thank our children, Aubree and Christian, for their patience and for encouraging me to finish this dissertation so that we can spend more time together. Thank God for providing for our family in all ways.

The original image pairs in figures 4.1 and 1.8 were provided by Kristin Branson and Neil McCurdy, respectively. Also, the sensors and software used to acquire data for this dissertation were borrowed from Neil McCurdy.

Chapter 1, in full, is being prepared for publication in collaboration with S. Belongie. I am the primary investigator and author of this paper.

Chapter 4, in part, namely section 4.1.1, is based on the paper "Covariance Propagation for Guided Matching" by B. Ochoa and S. Belongie [72]. I was the primary investigator and author of this paper.

VITA

1999	B. S., University of California, San Diego
2003	M. S., University of California, San Diego
2007	Ph. D., University of California, San Diego

PUBLICATIONS

B. Ochoa and S. Belongie, "Covariance Propagation for Guided Matching," Workshop on Statistical Methods in Multi-Image and Video Processing 2006.

J.R. Powell, S. Krotosky, B. Ochoa, D. Checkley, and P. Cosman, "Detection and Identification of Sardine Eggs at Sea Using a Machine Vision System," *Proceedings of the MTS/IEEE OCEANS*, 2003.

K.D. Moore, J.S. Jaffe, and B.L. Ochoa, "Development of a New Underwater Bathymetric Laser Imaging System: L-Bath," *Journal of Atmospheric and Oceanic Technology*, 17(8):1106–1117, August 2000.

P.P. Dang, B.L. Ochoa, and P.M. Chau, "Performance Analysis of Image Transmission Over Wireless Channels," *Proceedings of the SPIE Image and Video Communications and Processing*, 2000, pp. 908–916.

J.S. Jaffe, K.D. Moore, D. Zawada, B.L. Ochoa, and E. Zege, "Underwater Optical Imaging: New Hardware & Software," *Sea Technology*, 39(7):70–74, July 1998.

D.G. Zawada, J.S. Jaffe, A.M. Chekalyuk, and B.L. Ochoa, "Multispectral measurements of fluorescent biological pigments," AGU/ASLO Ocean Sciences Meeting 1998.

K.D. Moore, J. Jaffe, R. Currier, and B. Ochoa, "A new bathymetric laser imaging system," AGU/ASLO Ocean Sciences Meeting 1998.

ABSTRACT OF THE DISSERTATION

Precise Estimation of the Geoposition and Orientation of Ground-level Video Cameras from Multiple Sensors

by

Benjamin L. Ochoa
Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, & Control)

University of California San Diego, 2007

Professor Serge Belongie, Chair

Professor Nuno Vasconcelos, Co-Chair

This dissertation addresses the problem of precisely determining the geodetic position (geoposition) and orientation of multiple ground-level video cameras. Each video camera is calibrated and equipped with a Global Positioning System (GPS) receiver and compass-magnetometer. The GPS receiver measures the latitude, longitude, and height above mean sea level of the video camera and the orientation of the video camera is derived from data acquired by a compass-magnetometer, which measures the pitch, roll, and yaw of the camera. Additionally, features are tracked throughout the video acquired by the calibrated camera in order to measure the relative camera motion between successive video frames. Each of the measurements from this disparate set of sensors is first mapped such that they are all relative to a common Earth-centered, Earth-fixed Cartesian coordinate frame. The uncertainty of each measurement is also propagated through this mapping. The geoposition and orientation of each camera is independently estimated from all of its associated sensor measurements. The measurements and their associated uncertainties are input at different frequencies and are sometimes incomplete due to GPS dropouts, corrupt video frames, etc., yet the recursive estimation process uses these multiple measurements to reliably calculate the most probable geoposition and orientation with quantified uncertainty at each video frame.

Further, if multiple cameras are imaging the same region of a scene, cross-

camera feature correspondences are established using a combination of guided matching and robust feature comparison. The resulting independent observations of corresponding features contained in the scene are used to jointly estimate the maximum likelihood of the geoposition and orientation of all cameras imaging the same region of a scene for which feature correspondences have been established. This yields decreased relative errors between the cameras, resulting in more precise estimates of the geoposition and orientation of the cameras. This approach scales well and allows the video cameras to be located anywhere in the proximity of the Earth.

1

Overview

1.1 Introduction

We live in an age where almost anywhere you look, a camera is looking back at you. Whether it be in a convenience store, airport, or hotel lobby, while you are driving through an intersection or standing at a bank automated teller machine, or even while riding on a bus, surveillance cameras are there to capture and record the events as they unfold. These types of video cameras are usually mounted in fixed locations and are primarily used by government agencies and private security departments for continuous monitoring by human operators. Recently, with the technological advances and lowered costs of digital cameras, mobile cameras are pervasive as well. Especially with the success of the camera phone, many of which can capture video, mobile cameras have become ubiquitous [43]. Yes, mobile cameras are now everywhere, but where, precisely, are the cameras?

For several years, software applications have been able to combine images and video from cameras with Global Positioning System (GPS) receiver measurements synchronized with the acquisition of the images and video frames (e.g., [29]). More recently, camera phones have become GPS-enabled (e.g., [65]), providing an integrated, convenient, and inexpensive means of acquiring images and video with associated geodetic latitude and longitude. Such acquired data have been used in mapping applications such as navigation using visual landmarks, allowing users to view ground-level images of business storefronts up and down a street [1]. Although this data includes the geodetic position, or geoposition, of the camera at the time of acquisition, there is not an

associated measurement of the camera orientation. That is, the position of the camera is known, but the direction that it is pointing is unknown. Knowledge of the camera orientation expands the potential applications of this data from visual landmark-based navigation to, for example, georegistered urban 3D scene reconstruction from video [77]. However, the geoposition and orientation of the resulting 3D scene model is only as precise as the geoposition and orientation of the video camera.

Rather than develop a higher-level application, this dissertation addresses the more fundamental problem of improving the precision of geoposition and orientation estimates of one or more ground-level video cameras from measurements obtained from a GPS receiver, 3-axis orientation sensor, and calibrated video camera. An additional objective of this dissertation is to address this problem using consumer-grade sensors that are inexpensive and produced in large volumes.

This results of this work can be applied to any scenario where the accurate geoposition and orientation of multiple video cameras must be known in a timely manner. One example of this is video cameras attached to or carried by members of an emergency response team investigating the scene of an incident. Similar examples include soldiers wearing helmet-mounted cameras during battle, or a swarm of robots navigating a scene, perhaps autonomously. In either case, there is commonly an operations center that serves both as the central location for fusing the data acquired by the sensors and as the location where the controller of these assets resides. The controller may be a human, mentally fusing the data and using this information to control the cameras. Alternatively, the controller may be an autonomous system that moves the cameras based on a set of predefined rules. For example, the autonomous system may be a dedicated computer or the mother robot in a swarm. Alternatively, in systems based on a decentralized data fusion framework [67], it can be multiple computers or robots. In either situation, it is important that the geoposition and orientation are accurately known.

Previous work in the area is primarily focused on estimation of camera position and orientation relative to some local coordinate frame. Using video acquired by a calibrated camera, the translation and rotation of the camera can be reliably estimated relative to the camera position and orientation at the acquisition of the first video frame [90, 10, 53]. Alternative approaches to visual odometry have been successfully combined with other measurements such as an inertial navigation system (INS), which uses inertial detectors to determine the position, heading, and velocity of the system from

measurements of the 3D accelerations and rotations being applied to its inertial reference frame (e.g., [22]). Other approaches also include GPS measurements (e.g., [38, 68, 69]). Additionally, the robotics community has developed several approaches to simultaneous localization and mapping (SLAM) that include a video sensor (e.g., [16]).

For decades, the field of photogrammetry [59] has addressed the problem of precisely determining the geoposition and orientation of airborne and spaceborne cameras from acquired still images and initial position and orientation estimates. Recently, several approaches have been developed to extend these techniques to aerial cameras containing video imaging sensors; however, this community is still developing standards that specifically address cameras containing video imaging sensors [21]. Most promising is the application of photogrammetric techniques to the problem of video registration [87]. Similar to photogrammetric methods for still image georegistration, airborne video georegistration approaches utilize collateral data, including a 3D scene model (e.g., a digital elevation model (DEM)) and a high-resolution reference image acquired by a camera with precisely known model parameters [44, 98].

This dissertation aims to bring the geopositioning rigor of photogrammetry to multiple ground-level video cameras. Though the work presented in this dissertation does not make use of collateral data, the approach does use the combined measurements from a GPS receiver, relative orientation sensor, and calibrated video camera to sequentially estimate the precise geoposition and orientation of the cameras. The measurements from each of these sensors is mapped such that they are relative to an Earth-centered, Earth-fixed Cartesian coordinate frame. For each camera, the geoposition and orientation is sequentially estimated from all measurements. This estimation process incorporates the uncertainties associated with each of these measurements to calculate the most probable geoposition and orientation with quantified uncertainty. Further, if multiple cameras are imaging the same region of a scene, these independent observations of the features contained in the scene are used to further refine the geoposition and orientation of the cameras, resulting in reduced uncertainty of the estimates. This work enables other higher-level applications such as high-precision scene reconstruction from multi-camera video data.

This dissertation is structured as follows. This chapter provides a summary of the dissertation. Chapter 2 explains the sensor suite and details the mapping of sensor measurements to a common coordinate frame, including covariance propagation, and

is summarized in section 1.2. Chapter 3 discusses the approach used to sequentially estimate the geoposition and orientation of a single camera over time from the set of disparate sensor measurements and is summarized in section 1.3. Chapter 4 describes the extension of this work to include cross-camera measurements, in the case of multiple cameras imaging the same region of a scene, to further refine the geoposition and orientation of the cameras, and is summarized in section 1.4. Finally, experimental results and conclusions are given in section 1.5 and section 1.6.

Notation The following notation is used throughout this dissertation.

- Matrices are shown as uppercase letters in typewriter font, e.g., $\mathtt{M}$.
- Column vectors are shown as bold letters in Roman font, e.g., $\mathbf{v}$.
- Homogeneous coordinates in 2D are represented as 3-vectors and shown as lowercase letters, e.g., $\mathbf{x} = (x, y, w)^\top$. Additionally, normalized coordinates include a hat, e.g., $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{w})^\top$.
- Inhomogeneous coordinates in 2D are represented as 2-vectors and shown as lowercase letters with a tilde, e.g., $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})^\top = (x/w, y/w)^\top$, and may also include normalized coordinates, e.g., $\hat{\tilde{\mathbf{x}}} = (\hat{\tilde{x}}, \hat{\tilde{y}})^\top = (\hat{x}/\hat{w}, \hat{y}/\hat{w})^\top$.
- Homogeneous coordinates in 3D are represented as 4-vectors and shown as uppercase letters, e.g., $\mathbf{X} = (X, Y, Z, T)^\top$.
- Inhomogeneous coordinates in 3D are represented as 3-vectors and shown as uppercase letters with a tilde, e.g., $\widetilde{\mathbf{X}} = (\widetilde{X}, \widetilde{Y}, \widetilde{Z})^\top = (X/T, Y/T, Z/T)^\top$.
- If a capital letter is used to denote a matrix, then the vector denoted by the corresponding lower case letter is composed of the entries of the matrix by

$$\mathtt{A} \in \mathbb{R}^{m \times n} \Leftrightarrow \mathtt{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix}, \mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{pmatrix} \in \mathbb{R}^{mn}$$

  where $\mathbf{a}_i^\top \in \mathbb{R}^n$ is the ith row of $\mathtt{A}$ (i.e., $\mathbf{a} = \mathrm{vec}(\mathtt{A}^\top)$).

Any exceptions to the above notation will indicated at the time of their use.

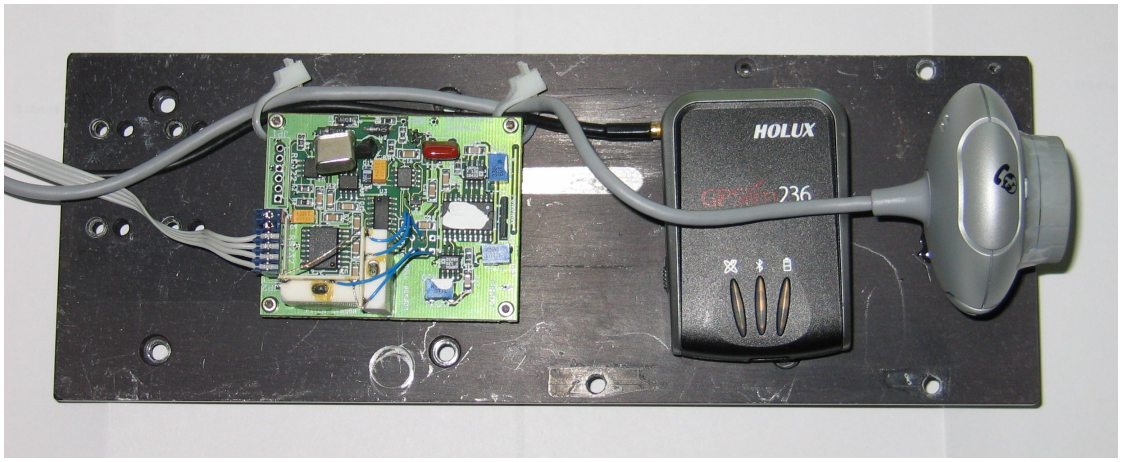


Figure 1.1: Sensors in the data acquisition system. From left to right are the tilt compensated compass-magnetometer, GPS receiver, and camera. The senors are rigidly mounted to 25 cm × 9 cm sheet of 0.5 cm-thick aluminum.

1.2 Sensors

Multiple sensors are required to precisely determine the geoposition and orientation of a video camera. In this work, the sensor system includes a video camera, GPS receiver, and 3-axis orientation sensor. Figure 1.1 shows the configuration of these sensors. The GPS receiver measures the latitude, longitude, and height above mean sea level of the video camera. The orientation of the video camera is derived from data acquired by a compass-magnetometer, which measures the pitch, roll, and yaw of the sensor platform. Additionally, camera motion estimates are determined from the video data of the calibrated camera. This section particularly details the mapping of the measurements from this set of disparate sensors to a common coordinate frame and the propagation of the uncertainty, in the form of covariance matrices, of the sensor measurements to this coordinate frame.

First-order nonlinear propagation of covariance is used throughout this dissertation and is briefly described here. Let $\mathbf{x} \in \mathbb{R}^n$ be a random vector with mean $\boldsymbol{\mu}_\mathbf{x}$ and covariance matrix $\Sigma_\mathbf{x}$, and let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a nonlinear function. Up to first-order approximation, $\mathbf{y} = f(\mathbf{x}) \approx f(\boldsymbol{\mu}_\mathbf{x}) + \mathtt{J}(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})$, where $\mathtt{J} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix $\partial f / \partial \mathbf{x}$ evaluated at $\boldsymbol{\mu}_\mathbf{x}$. If f is approximately affine in the region about the mean of the distribution, then this approximation is reasonable and the random vector $\mathbf{y} \in \mathbb{R}^m$ has mean $\boldsymbol{\mu}_\mathbf{y} \approx f(\boldsymbol{\mu}_\mathbf{x})$ and covariance $\Sigma_\mathbf{y} \approx \mathtt{J}\Sigma_\mathbf{x}\mathtt{J}^\top$.

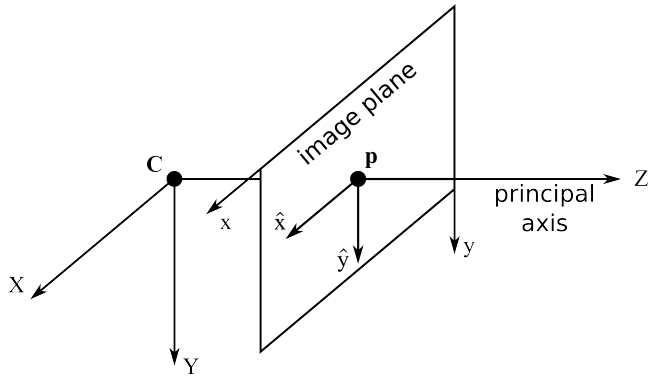


Figure 1.2: Relationship between coordinates in the camera coordinate frame, image coordinates, and normalized coordinates. **C** is the camera center and **p** is the principal point.

1.2.1 Video camera

The image formation process projects 3D world coordinates $\mathbf{X}$ to 2D image coordinates $\mathbf{x}$ that have been corrected for any lens distortion. This mapping is given by

$$\begin{aligned} \mathbf{x} &= \mathtt{K}[\mathtt{R} \mid \mathbf{t}]\mathbf{X} \\ \mathtt{K}^{-1}\mathbf{x} &= [\mathtt{R} \mid \mathbf{t}]\mathbf{X} \\ \hat{\mathbf{x}} &= \hat{\mathtt{P}}\mathbf{X} \end{aligned}$$

where $\mathtt{R}$ and $\mathbf{t}$ are the 3D rotation and translation, respectively, that transform coordinates in the world coordinate frame to coordinates in the camera coordinate frame. The upper triangular matrix $\mathtt{K}$ is called the camera calibration matrix and encompasses the intrinsic parameters of the camera in terms of pixel dimensions. $\hat{\mathtt{P}} = [\mathtt{R} \mid \mathbf{t}]$ is called the normalized camera projection matrix and represents a camera with an ideal lens that maps 3D coordinates $\mathbf{X}$ in the world coordinate frame to normalized 2D coordinates $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$ in the image plane. The relationship between 3D coordinates in the camera coordinate frame, image coordinates, and normalized coordinates is illustrated in figure 1.2.

In order to estimate the camera motion from video acquired from a calibrated camera, image coordinates must be mapped to normalized coordinates, $\tilde{\mathbf{x}} \mapsto \hat{\tilde{\mathbf{x}}}$. The covariance matrix $\Sigma_{\hat{\tilde{\mathbf{x}}}}$ associated with the normalized coordinates is calculated from the covariances of $\mathtt{K}$ and $\mathbf{x}$ as they propagate through the equation $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$. For clarity,

let $\mathtt{A} = \mathtt{K}^{-1}$, so

$$\begin{pmatrix} \hat{\tilde{x}} \\ \hat{\tilde{y}} \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \hat{\tilde{\mathbf{x}}} \\ 1 \end{pmatrix} = \mathtt{A} \begin{pmatrix} \tilde{\mathbf{x}} \\ 1 \end{pmatrix}$$

Using this notation, $\Sigma_{\hat{\tilde{\mathbf{x}}}} \approx \mathtt{J}_{\mathbf{a}} \Sigma_{\mathbf{a}} \mathtt{J}_{\mathbf{a}}^{\top} + \mathtt{J}_{\tilde{\mathbf{x}}} \Sigma_{\tilde{\mathbf{x}}} \mathtt{J}_{\tilde{\mathbf{x}}}^{\top}$, where $\Sigma_{\mathbf{a}} \approx \mathtt{J}_{\mathbf{k}} \Sigma_{\mathbf{k}} \mathtt{J}_{\mathbf{k}}^{\top}$, $\mathtt{J}_{\mathbf{a}} = \partial \hat{\tilde{\mathbf{x}}} / \partial \mathbf{a}$, $\mathtt{J}_{\tilde{\mathbf{x}}} = \partial \hat{\tilde{\mathbf{x}}} / \partial \tilde{\mathbf{x}}$, and $\mathtt{J}_{\mathbf{k}} = \partial \mathbf{a} / \partial \mathbf{k}$.

$\mathtt{K}$ and its associated covariance matrix $\Sigma_{\mathbf{k}}$ are usually determine during the camera calibration process. The covariance matrix $\Sigma_{\tilde{\mathbf{x}}}$ is a function of how the image coordinates $\tilde{\mathbf{x}}$ are measured, whether manually or automatically (e.g., by an autonomous feature detector). If the covariance of the measured coordinates is unknown, it is assumed that $\Sigma_{\tilde{\mathbf{x}}}$ is the identity matrix.

1.2.2 GPS receiver

The data acquisition system measures the height above mean sea level and geodetic latitude and longitude at a frequency of 1 Hz using a GPS receiver. For use in subsequent computations, these quantities are transformed to an Earth-centered, Earth-fixed Cartesian coordinate system as follows.

Geodetic coordinate transformation

Geodetic coordinates and their transformation have been extensively studied in the field of geospatial science [78, 79, 97]. Over 225 datums, each associated with one of 23 reference ellipsoids, are commonly used in mapping, charting, and geodesy [19]. Unique among these is the World Geodetic System (WGS) [20] because it is both a datum and a reference ellipsoid. As such, WGS provides the means for relating positions on various datums to an Earth-centered, Earth-fixed coordinate system.

The Earth Gravity Model 1996 (EGM96) [45] is a geopotential model of the Earth consisting of spherical harmonic coefficients complete to degree and order 360. This geopotential model is used as a geodetic reference to convert between EGM96 geoid height (i.e., height above mean sea level) to height above World Geodetic System

1984 (WGS84) ellipsoid, thereby correcting for any distance between the geoid and the mathematical reference ellipsoid as measured along the ellipsoidal normal.

Independent of both datum and reference ellipsoid is the reference frame of the coordinates. There are 33 common reference frames comprised of different coordinate systems, map projections, grids, and grid reference systems [18, 17, 89]. The general transformation of coordinates is performed using the approach described in [97], which is summarized as:

1. Convert the input coordinates from the input reference frame to the geodetic reference frame.
2. Shift the intermediate geodetic coordinates from the input datum to WGS84.
3. Convert between EGM96 geoid height to WGS84 ellipsoid height, if needed.
4. Shift the shifted intermediate WGS84 geodetic coordinates to the output datum.
5. Convert the shifted intermediate geodetic coordinates to the output coordinate reference frame.

For use in subsequent computations, the GPS receiver measurements of latitude, longitude, and height are transformed to geocentric coordinates. The relationship between the geodetic and geocentric coordinate systems is shown in figure 1.3. For GPS, the current underlying coordinate systems is WGS84—GPS receiver measurements are in WGS84 geodetic coordinates with EGM96 geoid height. WGS84 geocentric is also the principal coordinate frame used in the work presented in this dissertation.

GPS positioning uncertainty

The Global Positioning System (GPS) [64, 75, 76] is the most accurate worldwide navigation system developed to date. GPS was developed by the United States Department of Defense and presently consists of more than two dozen satellites, each with a highly accurate atomic clock, that orbit the Earth. Each satellite periodically transmits signals that report the satellite position and the transmission time. GPS receivers use these satellite messages to calculate the range to three or more satellites and then determine the position of the receiver using trilateration. However, GPS-derived positioning is not without error [56, 13, 5, 57, 74].

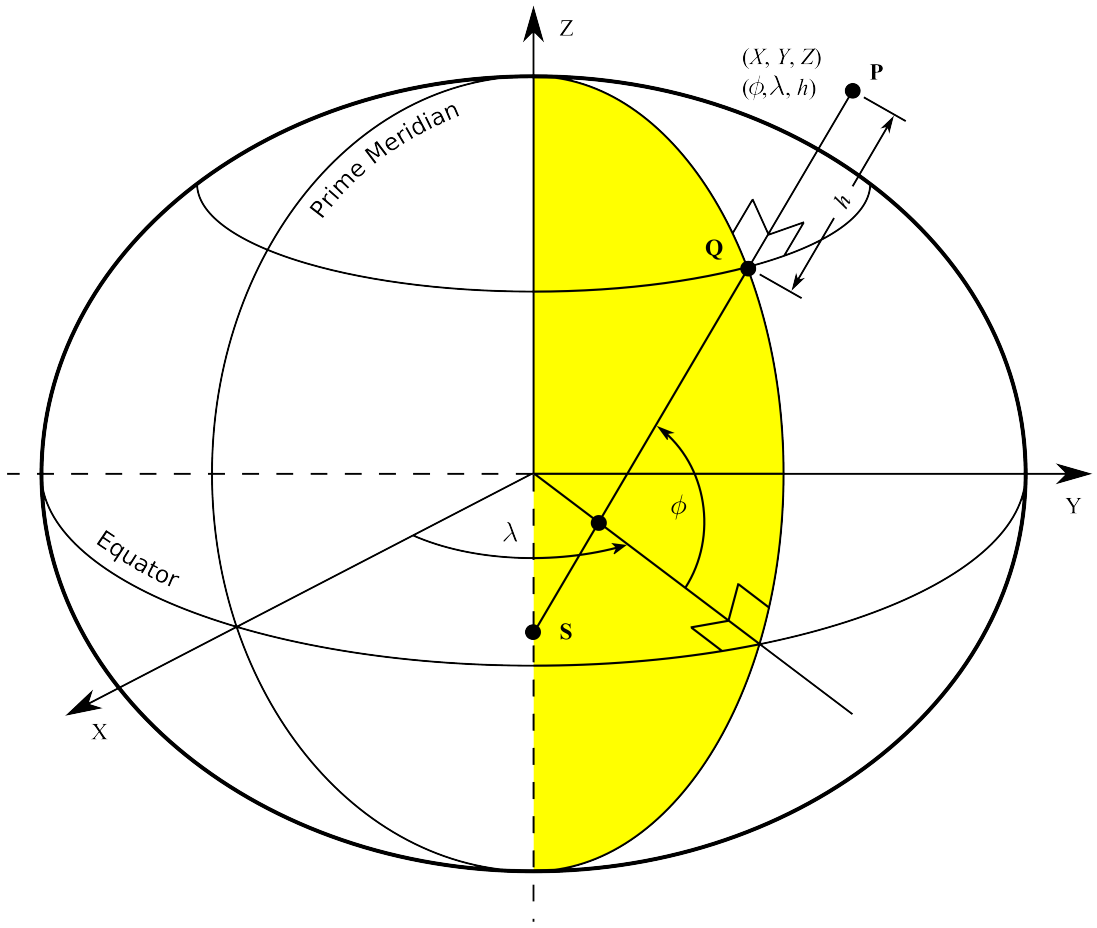


Figure 1.3: Relationship between geodetic and geocentric coordinates. The angle between the ellipsoidal normal $\overline{\mathbf{SP}}$ and the equatorial (XY-) plane defines the geodetic latitude ϕ of point **P**. The meridian containing **P** (in yellow) is defined as the half-plane containing the Z-axis and **P**. The angle between the prime meridian (XZ-) plane and the meridian containing **P** is the geodetic longitude λ of **P**. The distance from **Q** to **P** is the ellipsoidal height h.

GPS error is spread across the following classes: ephemeris (errors in the transmitted satellite position), satellite clock (errors in the satellite transmit time), ionosphere (errors due to ionospheric effects), troposphere (errors due to tropospheric effects), multipath (errors caused by multiple, reflected signals entering the receiver antenna), and receiver (errors due to thermal noise, software accuracy, an interchannel biases inherent in the receiver). The magnitude of each of these error sources is summarized in the standard GPS error shown in table 1.1. The magnitude of the error of the position along the ellipsoidal normal σ_{vertical} and in the plane orthogonal to the ellipsoidal normal $\sigma_{\text{horizontal}}$ are derived from the standard deviations of the standard error sources. The standard GPS errors assume the median geometric configuration of the satellites. The work presented in this dissertation uses greater values than those in the standard error table. Specifically, the covariance of the coordinates of the camera center in the WGS84 geocentric coordinate frame $\Sigma_{\widetilde{\mathbf{C}}} = \text{diag}\left(\sigma^2_{\text{GPS}}, \sigma^2_{\text{GPS}}, \sigma^2_{\text{GPS}}\right)$ where $\sigma_{\text{GPS}} = 33.3\,\text{meters}$. This more accurately models satellite geometry that is less optimal than the median satellite configuration.

1.2.3 3-axis orientation sensor

A tilt compensated compass-magnetometer is used to measure the rotation of the camera about 3 axes. This sensor characterizes 3D orientation by Euler angles in the so-called "XYZ" convention. In this convention, the rotation is given by pitch θ, roll ψ, and yaw ϕ angles. The angles define a rotated coordinate frame relative to a local (unrotated) coordinate frame with origin at the current position of the sensor, positive X-axis pointing north, positive Y-axis pointing west, and positive Z-axis pointing up, along the ellipsoidal normal. This is shown in figure 1.4.

The 3D orientation of the sensor is calculated as the composition of three rotations, a first rotation by an angle ϕ about the Z-axis, a second by an angle θ about the Y-axis, and a third by an angle ψ about the X-axis. The 3D rotation matrix $\mathtt{R}$ maps coordinates in the rotated coordinate frame to coordinates in the unrotated coordinate

Table 1.1: Standard GPS error model. This table lists the magnitude of several error sources at the median satellite configuration. Each source of error in the standard GPS error model is characterized by a bias (systematic) and random (white noise) error component. For each error source, the total standard deviation is calculated as the square root of the sum of the squared bias and squared random standard deviations. Similarly, the total standard deviation for all error sources is the square root of the sum of each of the squared error source standard deviations. Further, σ_{vertical} is the standard deviation of the position along the ellipsoidal normal and $\sigma_{\text{horizontal}}$ is the standard deviation of the position in the plane orthogonal to the ellipsoidal normal.

	Standard deviation σ (meters)		
Error source	Bias	Random	Total
Ephemeris	2.1	0.0	2.1
Satellite clock	2.0	0.7	2.1
Ionosphere	4.0	0.5	4.0
Troposphere	0.5	0.5	0.7
Multipath	1.0	1.0	1.4
Receiver	0.5	0.2	0.5
Total	5.1	1.4	5.3
σ_{vertical}			12.8
$\sigma_{\text{horizontal}}$			10.2

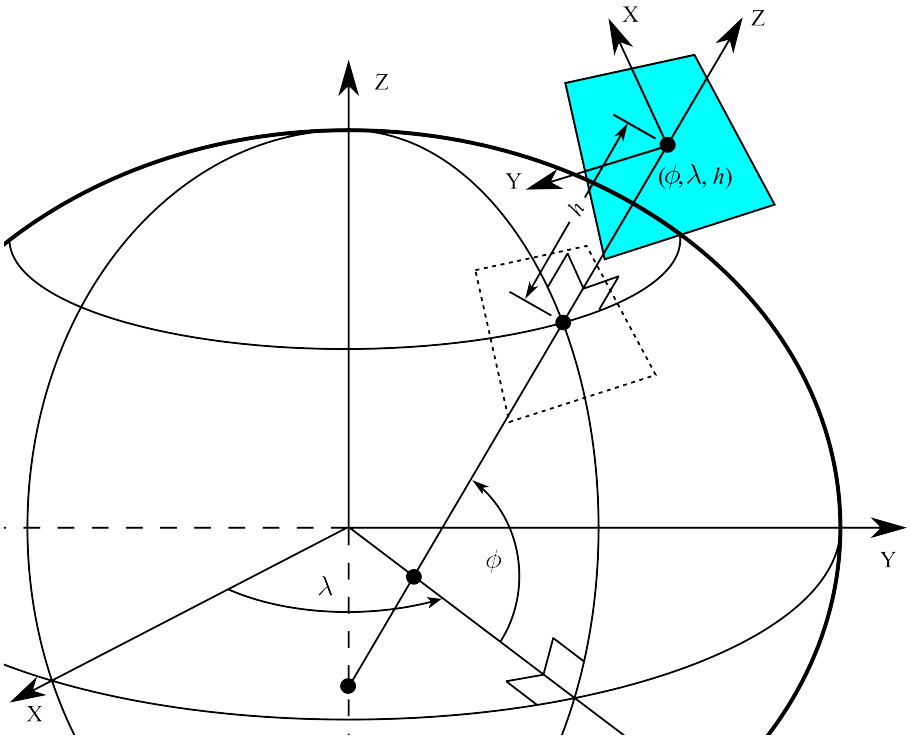


Figure 1.4: Pitch-roll-yaw unrotated local Cartesian coordinate frame. The local XY-plane (in blue) is parallel to the plane tangent to the surface of the reference ellipsoid at the geodetic latitude ϕ and longitude λ, and shifted along the ellipsoidal normal by the ellipsoid height h such that the local XY-plane contains $(\phi, \lambda, h)^\top$. The local X-axis points north, Y-axis points west, and Z-axis points up, along the ellipsoidal normal.

frame and is formed from the pitch, roll, and yaw measurements by

$$\mathtt{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathtt{R} = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ \sin\psi\sin\theta\cos\phi - \cos\psi\sin\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\theta & \cos\theta\sin\psi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\theta & \cos\theta\cos\psi \end{bmatrix} \tag{1.1}$$

Using covariance propagation, the covariance matrix associated with the rotation matrix $\mathtt{R}$ is given by $\Sigma_{\mathbf{r}} \approx \mathtt{J}_{\theta,\psi,\phi}\Sigma_{\theta,\psi,\phi}\mathtt{J}_{\theta,\psi,\phi}^\top$, where $\mathtt{J}_{\theta,\psi,\phi} = \partial\mathbf{r}/\partial(\theta,\psi,\phi)$. The pitch-roll-yaw measurement uncertainty $\Sigma_{\theta,\psi,\phi}$ is determined from specifications provided by the sensor manufacturer.

Rotation to camera coordinate frame and its uncertainty

As described above, the pitch, roll, and yaw measurements are relative to a local unrotated coordinate frame. Subsequent processing requires knowledge of the rotation that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame. This section describes the calculation of this rotation from GPS receiver measurements of the coordinates of the camera center in the WGS84 geocentric coordinate frame $\widetilde{\mathbf{C}}$ and its associated covariance $\Sigma_{\widetilde{\mathbf{C}}}$, and 3-axis orientation measurements of pitch θ, roll ψ, and yaw ϕ and the associated covariance $\Sigma_{\theta,\psi,\phi}$.

The rotation that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame is a composition of four rotations that transform coordinates through the following coordinate frames:

(a) WGS84 geocentric coordinate frame

(b) WGS84 local Cartesian coordinate frame

(c) Pitch-roll-yaw unrotated local Cartesian coordinate frame

(d) Pitch-roll-yaw rotated local Cartesian coordinate frame

(e) Camera coordinate frame

Of these, WGS84 local Cartesian has not yet been described. The WGS84 local Cartesian coordinate frame is similar to the pitch-roll-yaw unrotated local Cartesian coordinate frame shown in figure 1.4. The difference for WGS84 local Cartesian is that the positive X-axis points east and positive Y-axis points north. As with the pitch-roll-yaw unrotated local Cartesian coordinate frame, the Z-axis points up. Figure 1.5 shows the relationship between the 5 coordinate frames.

Most of the rotation matrices that transform coordinates from a given coordinate frame to the camera coordinate frame are straightforward to calculate. The rotation $\mathtt{R}_{d,e}$ from the pitch-roll-yaw rotated local Cartesian coordinate frame to the camera coordinate frame is given by

$$\mathtt{R}_{d,e} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

The rotation $\mathtt{R}_{c,d}$ from the pitch-roll-yaw unrotated local Cartesian coordinate frame to the pitch-roll-yaw rotated local Cartesian coordinate frame is calculated as

Figure 1.5: Relationship between 3D coordinate frames. Rendered along the top of the image are the coordinate frames relative to the image plane. From left to right are the camera coordinate frame, the pitch-roll-yaw rotated local Cartesian coordinate frame, pitch-roll-yaw unrotated local Cartesian coordinate frame, WGS84 local Cartesian coordinate frame, and WGS84 geocentric coordinate frame. For each coordinate frame, the X-axis is in red, Y-axis in green, and Z-axis in blue.

$\mathtt{R}_{c,d} = \mathtt{R}_{d,c}^{\top}$, where $\mathtt{R}_{d,c}$ is given by (1.1). The rotation $\mathtt{R}_{c,e}$ from the pitch-roll-yaw unrotated local Cartesian coordinate frame to the camera coordinate frame is the composition of the rotation $\mathtt{R}_{c,d}$ from the pitch-roll-yaw unrotated local Cartesian coordinate frame to the pitch-roll-yaw rotated local Cartesian coordinate frame and the rotation $\mathtt{R}_{d,e}$ from the pitch-roll-yaw rotated local Cartesian coordinate frame to the camera coordinate frame, $\mathtt{R}_{c,e} = \mathtt{R}_{d,e}\mathtt{R}_{c,d}$.

The rotation $\mathtt{R}_{b,c}$ from the WGS84 local Cartesian coordinate frame to the

pitch-roll-yaw unrotated local Cartesian coordinate frame is given by

$$\mathtt{R}_{b,c} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Similar to the composition of rotations above, the rotation $\mathtt{R}_{b,e}$ from the WGS84 local Cartesian coordinate frame to the camera coordinate frame is given by $\mathtt{R}_{b,e} = \mathtt{R}_{c,e}\mathtt{R}_{b,c}$.

The remaining rotation $\mathtt{R}_{a,b}$ from the WGS84 geocentric coordinate frame to the WGS84 local Cartesian coordinate frame is more involved, as it first requires establishment of the WGS84 local Cartesian coordinate frame. The origin of the WGS84 local Cartesian coordinate frame is at the camera center $\widetilde{\mathbf{C}}$, converted to WGS84 geodetic coordinates $(\phi, \lambda, h)^\top$. Next, the origin and each of the three standard basis vectors in the WGS84 geocentric coordinate frame are converted to WGS84 local Cartesian coordinates

$$\begin{aligned} (0,0,0)^\top &\mapsto \mathbf{X}_0^\top \\ (1,0,0)^\top &\mapsto \mathbf{X}_1^\top \\ (0,1,0)^\top &\mapsto \mathbf{X}_2^\top \\ (0,0,1)^\top &\mapsto \mathbf{X}_3^\top \end{aligned}$$

and $\mathtt{R}_{a,b}$ is given by $\mathtt{R}_{a,b} = [\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_2 - \mathbf{X}_0 \mid \mathbf{X}_3 - \mathbf{X}_0]$. Finally, the rotation $\mathtt{R}_{a,e}$ from the WGS84 geocentric coordinate frame to the camera coordinate frame is calculated by $\mathtt{R}_{a,e} = \mathtt{R}_{b,e}\mathtt{R}_{a,b}$. For a minimal parameterization of the final 3D rotation, the rotation matrix $\mathtt{R}_{a,e}$ is mapped to exponential coordinates $\boldsymbol{\omega} = \log(\mathtt{R}_{a,e})$, where $\mathtt{R}_{a,e} = \exp(\boldsymbol{\omega})$ is the inverse mapping. In exponential coordinates, a 3D rotation is parameterized by the 3-vector $\boldsymbol{\omega}$ that represents a rotation by an angle $\|\boldsymbol{\omega}\|$ about the axis $\boldsymbol{\omega}$ [66, 32, 53].

As detailed in this section, the rotation $\boldsymbol{\omega}$ that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame is dependent on the GPS receiver measurements of the coordinates of the camera center $\widetilde{\mathbf{C}}$ in the WGS84 geocentric coordinate frame and the 3-axis orientation measurements of pitch θ, roll ψ, and yaw ϕ. As such, in order to correctly model the uncertainty of $\boldsymbol{\omega}$, its joint covariance with $\widetilde{\mathbf{C}}$ must be calculated as follows.

$$\Sigma_{(\boldsymbol{\omega}^\top, \widetilde{\mathbf{C}}^\top)} \approx \mathtt{J}_{(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} \begin{bmatrix} \Sigma_{(\theta,\psi,\phi)} & \mathtt{0} \\ \mathtt{0} & \Sigma_{\widetilde{\mathbf{C}}} \end{bmatrix} \mathtt{J}^\top_{(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} \tag{1.2}$$

where

$$\mathtt{J}_{(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} = \frac{\partial(\boldsymbol{\omega}^\top, \widetilde{\mathbf{C}}^\top)}{\partial(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} = \begin{bmatrix} \frac{\partial \boldsymbol{\omega}}{\partial(\theta,\psi,\phi)} & \frac{\partial \boldsymbol{\omega}}{\partial \widetilde{\mathbf{C}}} \\ \mathtt{0} & \mathtt{I} \end{bmatrix}$$

1.3 Single Camera Estimation

For each video camera, initial measurements from the GPS receiver and 3-axis orientation sensor are used to derive measurements of the geoposition and orientation of the camera. Additionally, the calibrated video camera is used to measure the relative camera motion between successive frames. This section describes estimation of camera motion from video and the process of combining all of the derived sensor measurements such that the geoposition and orientation is precisely estimated at the time of each video frame acquisition.

The implemented estimation process is a recursive one that uses all of the measurements up to and including the current set of measurements to produce an estimate of the position and rotation of the camera relative to the WGS84 coordinate frame. Further, the sequential estimation process allows for both asynchronous measurements and unavailability of measurements. These properties are especially important to this application, as some measurements are not always available (e.g., GPS dropouts, corrupt video frames, etc.) and when they are available, different measurements arrive at different frequencies (e.g., the GPS receiver reports measurements at 1 Hz, while video frames are typically acquired at 30 Hz). The joint, sequential estimation process incorporates the uncertainties associated with each of these measurements to calculate the most probable geoposition and orientation with quantified uncertainty.

1.3.1 Sequential estimation of geoposition and orientation

Kalman filters are reliably used for estimating the motion of a calibrated camera from video acquired by the camera [90, 10, 53]. It is usual that this type of filter is applied to the problem of estimating the translation and rotation of the camera with respect to a relative coordinate system that is typically set to the coordinate frame of the camera at the time of the first video frame. This dissertation deals with the grander problem of estimating the camera position and orientation relative to an Earth-centered, Earth-fixed Cartesian coordinate system, namely the WGS84 geocentric coordinate frame. This section describes the parameters of the Kalman filter used to sequentially estimate the

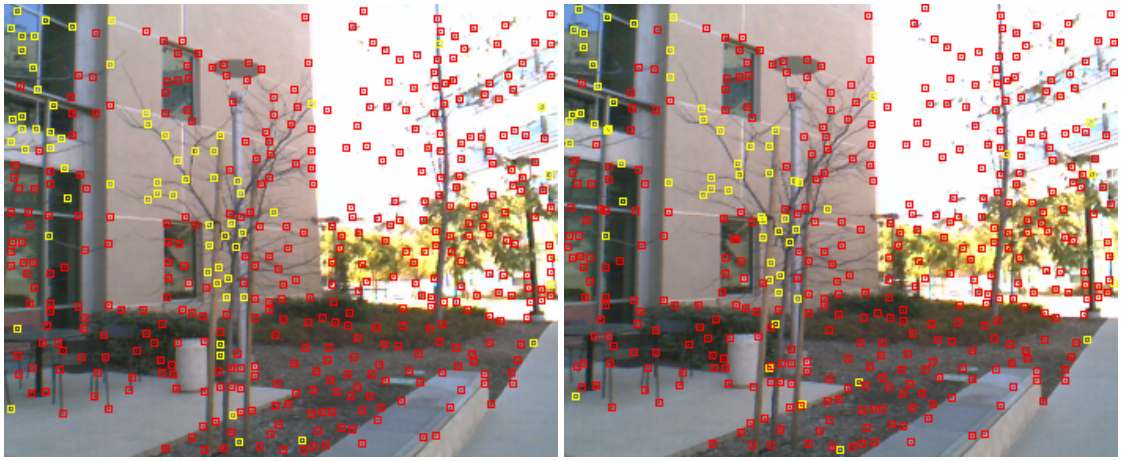
camera geoposition and orientation from video-derived measurements as well as from positional measurements derived from the GPS receiver and rotational measurements from the 3-axis orientation sensor. Advantages of such a Kalman filter are that it incorporates multiple, independent measurements and that it explicitly estimates the camera position and orientation in the WGS84 geocentric coordinate frame.

The mapping of measurements of camera position and rotation, and their uncertainty, from a GPS receiver and 3-axis orientation sensor, respectively, is detailed in section 1.2. Additionally, measurements of camera rotational velocity and direction of translational velocity may be derived by estimating the rotation $\boldsymbol{\omega}$ and translation $\mathbf{t}$ (to scale) of the camera from the previous video frame to the current one. In this case, the normalized cameras $\hat{\mathtt{P}}$ and $\hat{\mathtt{P}}'$ associated with the previous and current frames are given by $\hat{\mathtt{P}} = [\mathtt{I} \mid \mathbf{0}]$ and $\hat{\mathtt{P}}' = [\exp(\boldsymbol{\omega}) \mid \mathbf{t}]$, respectively. This is exactly the geometric relationship that is embodied by the essential matrix [47]. As such, estimation of the essential matrix between successive frames yields an estimate of the camera rotational velocity and direction of translational velocity. We estimate the essential matrix from a set of feature correspondences in normalized coordinates that have been tracked from the previous frame to the current frame as follows.

Motion estimation from video

For each video frame, good features to track are detected in the previous frame using the method described in [88]. A pyramidal implementation of the Lucas-Kanade feature tracker [52] then determines, for each feature, the translation from the previous frame to the current one. Multiresolution coarse-to-fine tracking allows for a larger window displacement from image to image while maintaining a smaller sized window. Figure 1.6 shows detected and tracked features between successive video frames.

Estimation of camera rotation and translation from images requires that image coordinates $\mathbf{x}$ are mapped to normalized coordinates $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$. The resulting set of point correspondences in normalized coordinates may contain incorrect correspondences [24] in the sense that they are inconsistent with the epipolar constraint of the essential matrix. This may be due, for example, to erroneous feature tracking or moving objects in the scene. Prior to estimation of the frame-to-frame camera rotation and translation, these incorrect correspondences are removed using the Random Sample Consensus (RANSAC) algorithm [23, 3]. The essential matrix is then estimated from the resulting set of inlier



a b

Figure 1.6: Feature detection and tracking. 446 features are detected in the previous frame (a) and tracked to the current frame (b). The parameters of the feature detector are a 5×5 window size and minor eigenvalue threshold of 10. Nonmaxima suppression is then applied such that there are no overlapping windows. Of the set of tracked features, 380 are inliers (in red) and 66 are outliers (in yellow), as determined using RANSAC.

correspondences.

The essential matrix $\mathtt{E}$ embodies the camera translation $\mathbf{t}$ and rotation $\boldsymbol{\omega}$, which have three degrees of freedom each. However, from a set of point correspondences, the essential matrix can only be determined to scale, i.e., the estimated essential matrix is a homogeneous entity. As such, it only has five degrees of freedom, which is insufficient to completely characterize $\mathbf{t}$ and $\boldsymbol{\omega}$. This constraint imposes that $\mathbf{t}$ can only be determined to scale, which indicates the direction of translation, but not its magnitude [32]. As with other homogeneous representations, it is convenient to constrain $\mathbf{t}$ such that $\|\mathbf{t}\| = 1$. Given the set of inlier correspondences from RANSAC, the Direct Linear Transformation (DLT) algorithm is used to initially estimate the essential matrix $\mathtt{E}$, which is then decomposed into a rotation $\boldsymbol{\omega}$ and translation $\mathbf{t}$ as described in [53].

Last, the maximum likelihood of the rotation $\boldsymbol{\omega}$ and unit translation $\mathbf{t}$ of the camera from the previous frame to the current one is estimated using bundle adjustment [93]. Bundle adjustment is a batch process that simultaneously adjusts the parameters of all of the cameras and the reconstructed 3D points $\mathbf{X}_1, \ldots, \mathbf{X}_n$ associated with the image feature correspondences such that the reprojection error is minimized. Initial estimates of the 3D points are determined in two steps. First, corrected correspondences that minimize the geometric error subject to the epipolar constraint are calculated for all inlier

feature correspondences. This is accomplished using the non-iterative, optimal method of [35]. Given the set of corrected correspondences, initial estimates of $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are estimated by triangulation using the DLT algorithm as described in [32].

From the initial rotation $\boldsymbol{\omega}$, translation $\mathbf{t}$, and 3D points $\mathbf{X}_1, \ldots, \mathbf{X}_n$, a specialized two-view bundle adjustment process is carried out using a sparse implementation of the Levenberg-Marquardt algorithm [46, 55, 34]. It is specialized in the sense that the parameters of the normalized camera $\hat{\mathtt{P}}^{(n-1)} = [\mathtt{I} \mid \mathbf{0}]$ associated with the previous frame $n-1$ are fixed to zero rotation and zero translation. Only the rotation $\boldsymbol{\omega}$ and translation $\mathbf{t}$ of the normalized camera $\hat{\mathtt{P}}^{(n)} = [\exp(\boldsymbol{\omega}) \mid \mathbf{t}]$ associated with the current frame n are adjusted. Additionally, $\mathbf{t}$ is constrained such that $\|\mathbf{t}\| = 1$ using the parameterization of the n-sphere [33, 71] throughout the bundle adjustment process. The Levenberg-Marquardt algorithm returns maximum likelihood estimates of $\boldsymbol{\omega}$ and $\mathbf{t}$ and the associated covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)}$.

Kalman filter

The extended Kalman filter (EKF) is an efficient recursive filter that estimates the state $\mathbf{x}$ of a dynamic system from a series of incomplete and noisy measurements [28]. At each time step n, the filter calculates the state estimate $\hat{\mathbf{x}}$ and its associated covariance $\mathtt{P}$ in two distinct phases: predict and update. The predict phase uses a model of the state transition from time step $n-1$ to time step n to calculate the a priori estimate of the state vector $\hat{\mathbf{x}}_n^-$ and a priori state error covariance estimate $\mathtt{P}_n^-$ at time step n. This is followed by the update phase, in which measurements and their associated covariances at time step n correct this prediction, yielding the a posteriori state estimate $\hat{\mathbf{x}}_n$ and a posteriori error covariance $\mathtt{P}_n$.

We now describe application of the EKF to the estimation of the geoposition and orientation of a camera at each video frame n. The filter estimates these parameters from measurements derived from video, the GPS receiver, and the 3-axis orientation sensor. It is assumed that the rotational velocity $\dot{\boldsymbol{\omega}}$ and positional velocity $\dot{\mathbf{C}}$ of the camera are constant between successive video frames, where each frame is a time step in the filter. Due to the high frame rate of typical video cameras, this is a reasonable assumption. Under this model, the Kalman filter state vector $\mathbf{x}$ is given by $\mathbf{x} = (\boldsymbol{\omega}^\top, \dot{\boldsymbol{\omega}}^\top, \mathbf{C}^\top, \dot{\mathbf{C}}^\top)^\top$, where $\mathbf{C}$ contains the coordinates of the camera center in the WGS84 geocentric coordinate frame, and $\boldsymbol{\omega}$ is the rotation from the WGS84 geocentric coordinate frame to the

camera coordinate frame. For clarity the tilde has been removed from the camera center.

For the predict phase of the filter, under a constant velocity state transition model, the entries in the a priori state estimate $\hat{\mathbf{x}}_n^-$ are given by

$$\begin{aligned}
\widehat{\boldsymbol{\omega}}_n^- &= \log(\exp(\widehat{\dot{\boldsymbol{\omega}}}_{n-1})\exp(\widehat{\boldsymbol{\omega}}_{n-1})) \\
\widehat{\dot{\boldsymbol{\omega}}}_n^- &= \widehat{\dot{\boldsymbol{\omega}}}_{n-1} \\
\widehat{\mathbf{C}}_n^- &= \widehat{\mathbf{C}}_{n-1} + \widehat{\dot{\mathbf{C}}}_{n-1} \\
\widehat{\dot{\mathbf{C}}}_n^- &= \widehat{\dot{\mathbf{C}}}_{n-1}
\end{aligned}$$

and the Jacobian matrix $\partial\hat{\mathbf{x}}_n^-/\partial\hat{\mathbf{x}}_{n-1}$ is given by

$$\frac{\partial\hat{\mathbf{x}}_n^-}{\partial\hat{\mathbf{x}}_{n-1}} = \begin{bmatrix}
\frac{\partial\widehat{\boldsymbol{\omega}}_n^-}{\partial\widehat{\boldsymbol{\omega}}_{n-1}} & \frac{\partial\widehat{\boldsymbol{\omega}}_n^-}{\partial\widehat{\dot{\boldsymbol{\omega}}}_{n-1}} & \mathtt{0} & \mathtt{0} \\
\mathtt{0} & \mathtt{I} & \mathtt{0} & \mathtt{0} \\
\mathtt{0} & \mathtt{0} & \frac{\partial\widehat{\mathbf{C}}_n^-}{\partial\widehat{\mathbf{C}}_{n-1}} & \frac{\partial\widehat{\mathbf{C}}_n^-}{\partial\widehat{\dot{\mathbf{C}}}_{n-1}} \\
\mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{I}
\end{bmatrix}$$

where

$$\frac{\partial\widehat{\boldsymbol{\omega}}_n^-}{\partial\widehat{\boldsymbol{\omega}}_{n-1}} = \frac{\partial\widehat{\boldsymbol{\omega}}_n^-}{\partial\hat{\mathbf{r}}_n^-}\frac{\partial\hat{\mathbf{r}}_n^-}{\partial\hat{\mathbf{r}}_{n-1}}\frac{\partial\hat{\mathbf{r}}_{n-1}}{\partial\widehat{\boldsymbol{\omega}}_{n-1}} \quad \text{and} \quad \frac{\partial\widehat{\boldsymbol{\omega}}_n^-}{\partial\widehat{\dot{\boldsymbol{\omega}}}_{n-1}} = \frac{\partial\widehat{\boldsymbol{\omega}}_n^-}{\partial\hat{\mathbf{r}}_n^-}\frac{\partial\hat{\mathbf{r}}_n^-}{\partial\hat{\dot{\mathbf{r}}}_{n-1}}\frac{\partial\hat{\dot{\mathbf{r}}}_{n-1}}{\partial\widehat{\dot{\boldsymbol{\omega}}}_{n-1}}$$

In the update phase of the filter, the a priori state estimate $\hat{\mathbf{x}}_n^-$ is corrected by three potential measurements: the camera position derived from the GPS receiver measurements, camera rotation derived from the 3-axis orientation measurements, or camera rotational velocity derived from video. Note that camera positional velocity measurements derived from video are not used. This is due to the fact that the camera may not translate between successive frames or translate by such a small magnitude, that the translation estimate is erroneous due to noise. However, it has been shown through experimentation that camera rotation is correctly estimated, despite an incorrect translation estimate [90, 53].

When the GPS receiver reports a new measurement, it is immediately converted to WGS84 geocentric coordinates $\mathbf{C}$ with associated covariance matrix $\Sigma_{\mathbf{C}}$ as detailed in section 1.2.2. If the 3-axis orientation sensor also reports a measurement in the same time step, then the origin of the camera coordinate frame is set to $\mathbf{C}$. As such, the calculated rotation $\boldsymbol{\omega}$ that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame is correlated to $\mathbf{C}$ and the covariances of $(\theta, \psi, \phi)^\top$ and $\mathbf{C}$ are jointly propagated to the covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{C}^\top)}$ given by (1.2).

In this case, the a priori state estimate is corrected by the measurement $(\boldsymbol{\omega}^\top, \mathbf{C}^\top)^\top$ with associated covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{C}^\top)}$. Otherwise, if a 3-axis orientation sensor measurement is not reported, then only the GPS derived measurement of the camera center is used to update the a priori state estimate using the measurement $\mathbf{C}$ with covariance $\Sigma_{\mathbf{C}}$.

In the case of a reported 3-axis orientation sensor measurement and not a GPS receiver measurement, the origin of the camera coordinate frame is set to the a priori estimate of the camera center $\widehat{\mathbf{C}}_n^-$ where the associated covariance $\Sigma_{\widehat{\mathbf{C}}_n^-}$ is the 3×3 block on the diagonal of the a priori state error covariance estimate $\mathtt{P}_n^-$ corresponding to $\widehat{\mathbf{C}}_n^-$. Similar to above, the rotation $\boldsymbol{\omega}$ calculated from the pitch, roll, and yaw $(\theta, \psi, \phi)^\top$ measurements is correlated to $\widehat{\mathbf{C}}_n^-$, and the covariances of $(\theta, \psi, \phi)^\top$ and $\widehat{\mathbf{C}}_n^-$ are jointly propagated to the covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \widehat{\mathbf{C}}_n^{-\top})}$. However, since $\widehat{\mathbf{C}}_n^-$ is not a measurement, it is not used to correct the a priori state estimate $\hat{\mathbf{x}}_n^-$ (i.e., it is not used to correct itself). Only the derived measurement of the rotation is used to update the a priori state estimate. Specifically, the state estimate is updated by the measurement $\boldsymbol{\omega}$ with covariance $\Sigma_{\boldsymbol{\omega}}$.

The last potential measurement is that of the camera rotational velocity $\dot{\boldsymbol{\omega}}$. A measure of the rotational velocity is the estimate of the rotation of the camera from the previous frame $n-1$ to the current one n as estimated by the previously described two-view bundle adjustment process. The state estimate is updated by $\dot{\boldsymbol{\omega}}$ and its associated covariance $\Sigma_{\dot{\boldsymbol{\omega}}}$.

The filter uses the measurements provided by this combination of sensors to sequentially estimate the geoposition and orientation of the camera relative the WGS84 geocentric coordinate frame. The measurements and their associated uncertainties are input at different frequencies and are sometimes incomplete due to GPS dropouts, corrupt video frames, etc., yet the EKF uses these multiple measurements to reliably estimate the geoposition and orientation and their uncertainty at each video frame.

1.4 Multiple Camera Estimation

This section extends the work of the previous one from independent estimation of the geoposition and orientation of each camera to that of jointly estimating the geoposition and orientation of multiple cameras at instances when this is possible. Specifically,

in the case of multiple cameras imaging the same region of a scene, these independent observations of the features in the scene are used to further refine the geoposition and orientation of the cameras, provided that feature correspondences are established between the images acquired from different cameras. The approaches used for feature detection and matching are described, as well as the process of jointly estimating the maximum likelihood of the geoposition and orientation of all cameras imaging the same region of a scene for which feature correspondences have been established.

1.4.1 Feature detection and matching

Recent work has shown that distinct image features that are invariant to viewpoint and illumination changes can be reliably detected [51, 63]. These types of changes are locally modeled as an affinity or similarity (translation, rotation, and scale). Examples of such feature detectors include ones based on affine normalization and Hessian points [61, 82], the Maximally Stable Extremal Region (MSER) detector [58], detectors based on edges and intensity extrema [96, 95], one that detects salient regions [40], and the Scale Invariant Feature Transform (SIFT) detector [49, 50, 51]. An affinity is sufficient to locally model geometric distortions arising from viewpoint changes provided that the local neighborhood about the scene feature can be approximated by a plane. Although a similarity does not model skew, it has been shown to perform well in similar applications, such as robotics [83, 84]. It is also assumed that photometric deformations can be modeled by a linear transformation of the local intensities. In this dissertation, image features are detected using the SIFT detector. Examples of regions detected by the SIFT detector are shown in figure 1.7.

For each detected region, a local description of the intensity pattern within the region is calculated. The feature matching process, described later, uses these local descriptors to determine the similarity between different features. A recent comparison of local descriptors [62] indicates that SIFT descriptors, each typically a 128-dimensional vector representing a local image region sampled relative to its scale-space coordinate frame, are superior to other descriptors. Further, the vector is organized such that the Euclidean distance between any two SIFT descriptor vectors is a measure of the similarity between the SIFT features described by the vectors, i.e., smaller distances are more similar. The work presented here uses the SIFT reference implementation [48] for both feature detection and calculation of the local descriptor.

Figure 1.7: Features for matching. SIFT features (in yellow and red) detected in images acquired by two different cameras. 127 and 101 features were detected in the left and right images, respectively. A combination of guided matching, SIFT descriptor comparison, and robust modeling fitting is used to determine the set of feature correspondences between the two images—29 of these feature points are inliers (in red) and the remaining detected points are outliers (in yellow).

The remainder of this section addresses the problem of matching the detected features across images acquired from different cameras that are imaging the same region of a scene. Focus is given to determination of the region in an image to search for a corresponding feature—the guided matching problem. Other components of the matching process are feature comparison to establish an initial set of correspondences followed by robust outlier rejection.

Guided matching

When it has been determined that multiple cameras are imaging the same region of a scene, the detected SIFT features in each of the images is robustly matched. The feature matching process first establishes a set of putative correspondences between SIFT features that have been detected in each of images acquired by the cameras. Putative correspondences are computed using a combination of guided matching using covariance propagation [72] and the comparison of SIFT descriptors. RANSAC is then applied to the set of putative feature correspondences to determine the set of inlier correspondences. Presently, the matching process is tailored to work on images acquired from pairs of normalized cameras $\hat{\mathtt{P}}$ and $\hat{\mathtt{P}}'$. In the case of three or more cameras imaging the same region of the scene, all possible image pairs are processed, and the results merged.

Guided matching is performed in the space of normalized coordinates, so the matching process first converts the detected SIFT features in the image acquired by the first normalized camera $\hat{\mathtt{P}}$ from image coordinates to normalized coordinates $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$ with uncertainty propagation. Additionally, the current a posteriori estimate of the Kalman filter state vector $(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)^\top$ of the first camera is mapped to a vector $(\boldsymbol{\omega}^\top, \mathbf{t}^\top)^\top$ containing the parameters of the normalized camera $\hat{\mathtt{P}} = [\mathtt{R} \mid \mathbf{t}] = [\exp(\boldsymbol{\omega}) \mid \mathbf{t}]$, where $\mathtt{R} = \exp(\boldsymbol{\omega})$ and $\mathbf{t} = -\mathtt{R}\widetilde{\mathbf{C}}$. The covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)}$ associated with the vector $(\boldsymbol{\omega}^\top, \mathbf{t}^\top)^\top$ is calculated by

$$\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)} \approx \mathtt{J}_{(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)} \Sigma_{(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)} \mathtt{J}^\top_{(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)}$$

where $\Sigma_{(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)}$ is the a posteriori state error covariance estimate, and the Jacobian matrix $\mathtt{J}_{(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)}$ is given by

$$\mathtt{J}_{(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)} = \frac{\partial(\boldsymbol{\omega}^\top, \mathbf{t}^\top)}{\partial(\widehat{\boldsymbol{\omega}}^\top, \widehat{\dot{\boldsymbol{\omega}}}^\top, \widehat{\mathbf{C}}^\top, \widehat{\dot{\mathbf{C}}}^\top)} = \begin{bmatrix} \mathtt{I} & \mathtt{0} & \mathtt{0} & \mathtt{0} \\ \frac{\partial \mathbf{t}}{\boldsymbol{\omega}} & \mathtt{0} & \frac{\partial \mathbf{t}}{\partial \widetilde{\mathbf{C}}} & \mathtt{0} \end{bmatrix}$$

Similarly, the a posteriori state estimate $(\widehat{\boldsymbol{\omega}}'^\top, \widehat{\dot{\boldsymbol{\omega}}}'^\top, \widehat{\mathbf{C}}'^\top, \widehat{\dot{\mathbf{C}}}'^\top)^\top$ of the second camera is mapped to a vector $(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)^\top$ containing the parameters of the second normalized camera $\hat{\mathtt{P}}' = [\mathtt{R}' \mid \mathbf{t}'] = [\exp(\boldsymbol{\omega}') \mid \mathbf{t}']$,

For use in guided matching between images acquired by every combination of camera pairs, the essential matrix $\mathtt{E}$ from a given pair of general normalized cameras $\hat{\mathtt{P}} = [\mathtt{R} \mid \mathbf{t}]$ and $\hat{\mathtt{P}}' = [\mathtt{R}' \mid \mathbf{t}']$ is given by

$$\mathtt{E} = \frac{[\mathbf{t}' - \mathtt{R}'\mathtt{R}^\top\mathbf{t}]_\times \mathtt{R}'\mathtt{R}^\top}{\|\mathbf{t}' - \mathtt{R}'\mathtt{R}^\top\mathbf{t}]_\times \mathtt{R}'\mathtt{R}^\top\|}$$

where $\mathtt{R} = \exp(\boldsymbol{\omega})$ and $\mathtt{R}' = \exp(\boldsymbol{\omega}')$. The covariance matrix $\Sigma_{\mathbf{e}}$ associated with $\mathtt{E}$ is calculated as

$$\Sigma_{\mathbf{e}} \approx \mathtt{J}_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)} \Sigma_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)} \mathtt{J}^\top_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)} + \mathtt{J}_{(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)} \Sigma_{(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)} \mathtt{J}^\top_{(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)}$$

where $\mathtt{J}_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)} = \partial \mathbf{e} / \partial(\boldsymbol{\omega}^\top, \mathbf{t}^\top)$ and $\mathtt{J}_{(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)} = \partial \mathbf{e} / \partial(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)$.

The resulting essential matrix $\mathtt{E}$ and covariance $\Sigma_{\mathbf{e}}$ are used to calculate search regions in image 2. A search region in image 2 corresponding to a detected SIFT feature in image 1 is determined by the mapping

$$\hat{\boldsymbol{\ell}}' = \frac{\mathtt{E}\hat{\mathbf{x}}}{\|\mathtt{E}\hat{\mathbf{x}}\|}$$

where $\hat{\mathbf{x}}$ is the point in normalized coordinates in image 1 and $\hat{\boldsymbol{\ell}}'$ is the line in normalized coordinates in image 2 with covariance $\Sigma_{\hat{\boldsymbol{\ell}}'} \approx \mathtt{J}_{\mathbf{e}}\Sigma_{\mathbf{e}}\mathtt{J}_{\mathbf{e}}^{\top} + \mathtt{J}_{\hat{\mathbf{x}}}\Sigma_{\hat{\mathbf{x}}}\mathtt{J}_{\hat{\mathbf{x}}}^{\top}$, where $\mathtt{J}_{\mathbf{e}} = \partial\hat{\boldsymbol{\ell}}'/\partial\mathbf{e}$ and $\mathtt{J}_{\hat{\mathbf{x}}} = \partial\hat{\boldsymbol{\ell}}'/\partial\hat{\mathbf{x}}$.

As described in [72], the set of equal-likelihood lines in the distribution of a random homogeneous line $\boldsymbol{\ell}$ with mean $\boldsymbol{\mu}_{\boldsymbol{\ell}}$ and covariance $\Sigma_{\boldsymbol{\ell}}$ satisfies $(\boldsymbol{\ell}-\boldsymbol{\mu}_{\boldsymbol{\ell}})^{\top}\Sigma_{\boldsymbol{\ell}}^{+}(\boldsymbol{\ell}-\boldsymbol{\mu}_{\boldsymbol{\ell}}) = k^2$, where k^2 is the inverse of the chi-square cumulative distribution function with 2 degrees of freedom and probability α, and $\Sigma_{\boldsymbol{\ell}}^{+}$ is the pseudo-inverse of the covariance matrix $\Sigma_{\boldsymbol{\ell}}$ with rank 2. The set of lines form the homogeneous dual conic $\mathtt{C}^* = \left[\boldsymbol{\mu}_{\boldsymbol{\ell}}\boldsymbol{\mu}_{\boldsymbol{\ell}}^{\top} - k^2\Sigma_{\boldsymbol{\ell}}\right]^{-1}$, which is the the adjoint of the matrix $\mathtt{C}$. For a non-singular symmetric matrix $\mathtt{C} \sim (\mathtt{C}^*)^{-1}$, therefore the conic that forms the envelope of lines is given by $\mathtt{C} = \boldsymbol{\mu}_{\boldsymbol{\ell}}\boldsymbol{\mu}_{\boldsymbol{\ell}}^{\top} - k^2\Sigma_{\boldsymbol{\ell}}$. This conic is a hyperbola with branches symmetric about $\boldsymbol{\mu}_{\boldsymbol{\ell}}$ as illustrated in figure 1.8.

In the present case of the line $\hat{\boldsymbol{\ell}}'$ in normalized coordinates in image 2 and its associated covariance $\Sigma_{\hat{\boldsymbol{\ell}}'}$, the conic $\hat{\mathtt{C}}'$ in normalized coordinates in image 2 that defines the search region is given by $\hat{\mathtt{C}}' = \hat{\boldsymbol{\ell}}'\hat{\boldsymbol{\ell}}'^{\top} - k^2\Sigma_{\hat{\boldsymbol{\ell}}'}$. An arbitrary point $\hat{\mathbf{x}}'$ lies inside the search region if $\hat{\mathbf{x}}'^{\top}\hat{\mathtt{C}}'\hat{\mathbf{x}}'$ has the same sign as $\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}'}'^{\top}\hat{\mathtt{C}}'\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}'}'$, where $\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}'}'$ is any point that lies on the line $\hat{\boldsymbol{\ell}}'$. Two points $\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}_1'}'$ and $\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}_2'}'$ on the line $\hat{\boldsymbol{\ell}}'$ may be determined by $\hat{\boldsymbol{\ell}}'^{\top}[\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}_1'}' \mid \hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}_2'}'] = 0$, where the matrix $[\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}_1'}' \mid \hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}_2'}']$ is the null space of $\hat{\boldsymbol{\ell}}'^{\top}$. One of these points can be used to determine the sign of $\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}'}'^{\top}\hat{\mathtt{C}}'\hat{\mathbf{x}}_{\hat{\boldsymbol{\ell}}'}'$.

Detected SIFT features in image 2 located within the search region meet the geometric criteria for potentially corresponding to $\hat{\mathbf{x}}$, regardless of how similar they are to the feature in image 1. Comparison of the descriptors is used to determine which of these features, if any, is similar to the feature in image 1. The matching process calculates how similar the potential corresponding features are as well as how unique the potential match is. For a detected SIFT feature in image 1, the matching process measures the Euclidean distance between its associated SIFT descriptor vector and all descriptor vectors contained in its corresponding search region in image 2, storing the distances to its nearest and second nearest neighbors, i.e., the smallest and second smallest Euclidean distances, respectively. The ratio of the smallest distance to the second smallest distance is a measure of how ambiguous the match is [51]. The lower the ratio, the less ambiguous the match. Thresholding on this ratio is an effective method for removing ambiguous matches.

Last, RANSAC is applied to the resulting set of putative correspondences to determine the subset of correspondences that are consistent with the essential matrix.

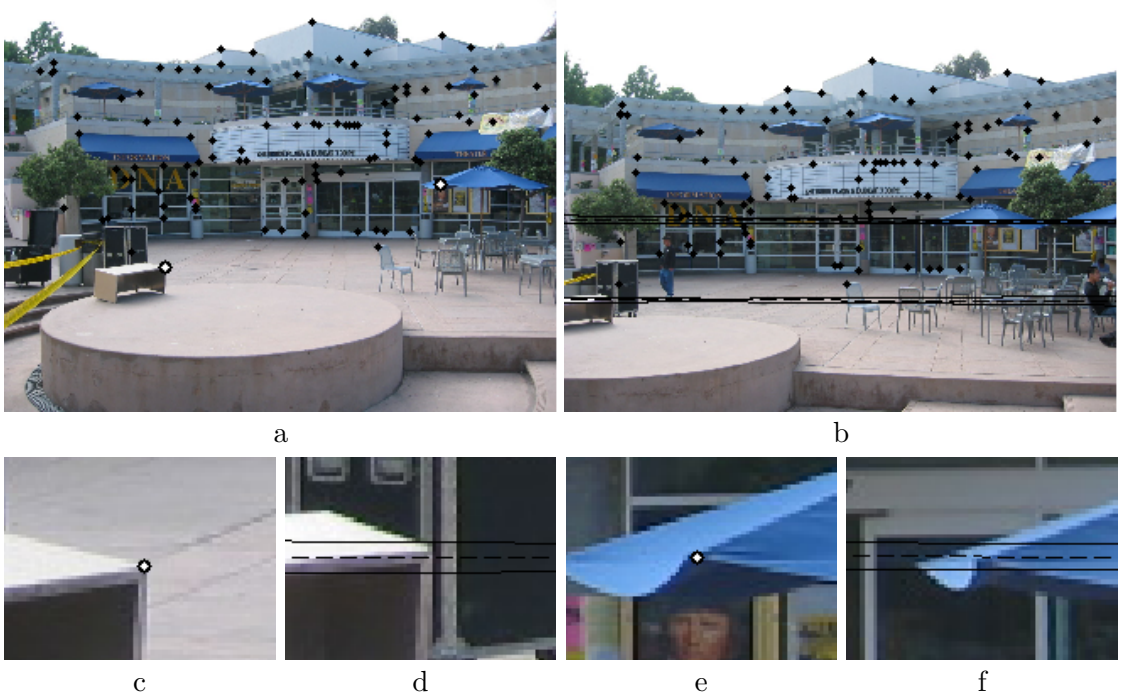


Figure 1.8: Point-to-line mapping under a fundamental matrix. (a) (b) The left and right images with corresponding points used in bundle adjustment (in black). There are 101 point correspondences. Two additional points have been selected in the left image (in white) and mapped to lines in the right image (in dashed black). The uncertainty hyperbolas associated with the mapped lines are contained in the right image (in black). The hyperbolas correspond to a probability of 99%. (c) The left image zoomed in on the first selected point. (d) The right image zoomed in on the corresponding first mapped line. (e) The left image zoomed in on the second selected point. (f) The right image zoomed in on the corresponding second mapped line. Note that the mapped lines miss the corresponding points, but that the corresponding points are within the uncertainty bounds.

Figure 1.7 shows example results of matching features across images acquired from different cameras using the procedure described in this section.

1.4.2 Joint estimation of geoposition and orientation

In the absence of cross-camera information, the system is essentially a set of independent Kalman filters, each estimating the geoposition and orientation of its respectively camera from its own sensor measurements. However, when multiple cameras image the same region of a scene, there is the potential to use cross-camera measurements to further improve estimates of all cameras that observe the same scene features. Matching features between images acquired by different cameras introduces the sharing of information across the cameras. There are multiple techniques for combining this additional information in order to improve the estimates of the geopositions and orientations of the cameras. These approaches range from a single Kalman filter with a state vector containing all of the parameters for all of the cameras to, for example, a decentralized data fusion framework [67]. Most of these approaches have been developed to mitigate issues that arise when the number of cameras significantly increases, for example, from tens of cameras to tens of thousands. Those techniques that do scale to a large number of cameras must often sacrifice some information for the ability to scale. The approached developed in this work falls into this category.

The method used in this work is a hybrid one. Each camera continues to independently estimate its geoposition and orientation as described in the section 1.3. However, when two or more cameras image the same region of a scene and feature correspondences are established, a separate, independent bundle adjustment process will simultaneously estimate the geoposition and orientation of these cameras from the set of feature correspondences between their images and current estimates of geoposition and orientation of the cameras. The results of this separate bundle adjustment process are then input to each of the Kalman filters as simply another correlated measurement of position and orientation. After the measurement update, the Kalman filters return to independent processing. The information that is lost by using this approach is the cross-camera covariance information resulting from bundle adjustment.

Bundle adjustment can reliably estimate positions and orientations of hundreds of cameras simultaneously [93]. Under typical operating conditions of this system, multiple bundle adjustment processes will be executing, each adjusting perhaps tens of

cameras, which is easily handled. With this in mind, the loss of cross-camera covariance information is considered an acceptable loss. The advantage of bundle adjustment is, through the use of cross-camera image feature correspondences, it allows the cameras to transfer their accuracy to each other by jointly estimating the geoposition and orientation of the cameras.

A sparse implementation of the Levenberg-Marquardt algorithm [34] is used to perform bundle adjustment, allowing computationally efficient adjustment the geoposition and orientation of m cameras as follows. For clarity, the hat notation is removed from the normalized image coordinates in the measurement vector $\mathbf{X}$. The initial estimate of the parameter vector is

$$\widehat{\mathbf{P}} = (\boldsymbol{\omega}^{(1)\top}, \widetilde{\mathbf{C}}^{(1)\top}, \ldots, \boldsymbol{\omega}^{(m)\top}, \widetilde{\mathbf{C}}^{(m)\top}, \widetilde{\mathbf{X}}_1^{\top}, \ldots, \widetilde{\mathbf{X}}_n^{\top})^{\top}$$

where the jth camera rotation $\boldsymbol{\omega}^{(j)}$ and center $\widetilde{\mathbf{C}}^{(j)}$ are initialized to the values in the current Kalman filter state estimate associated with the jth camera imaging the scene, and the 3D points $\widetilde{\mathbf{X}}_1, \ldots, \widetilde{\mathbf{X}}_n$ are initialized by triangulation using the DLT algorithm. The measurement vector $\mathbf{X}$ is given by

$$\mathbf{X} = (\boldsymbol{\omega}^{(1)\top}, \widetilde{\mathbf{C}}^{(1)\top}, \tilde{\mathbf{x}}_1^{(1)\top}, \ldots, \tilde{\mathbf{x}}_n^{(1)\top}, \ldots, \boldsymbol{\omega}^{(m)\top}, \widetilde{\mathbf{C}}^{(m)\top}, \tilde{\mathbf{x}}_1^{(m)\top}, \ldots, \tilde{\mathbf{x}}_n^{(m)\top})^{\top}$$

where $\tilde{\mathbf{x}}_i^{(j)}$ is the ith corresponding point in normalized coordinates in the jth camera. Notice that the above measurement vector also includes the current Kalman filter state estimates of the rotations and translations of the cameras. Inclusion of the rotations and translations in the measurement vector prevents their counterparts in the parameter vector from being adjusted outside of the uncertainty bounds of the current state estimate.

After bundle adjustment, the resulting rotations and translations are extracted from the final estimate of the parameter vector $\widehat{\mathbf{P}}$ and their covariances retrieved. Measurement updates of rotation and translation are issued to each Kalman filter associated with an adjusted camera. The result is decreased relative error between the cameras, resulting in more precise estimates of the geoposition and orientation of the cameras.

1.5 Experimental results

The approach developed in this dissertation has been experimentally validated using the following method. Data was acquired using the system shown in figure 1.1.

The data acquisition system consists of a collection of inexpensive, consumer-grade sensors: a calibrated Microsoft LifeCam VX-6000 video camera [60], Holux GPSlim236 GPS receiver [37], and Advanced Orientation Systems EZ-COMPASS-3 tilt compensated compass-magnetometer [2]. The imaging components of this camera are a 72° diagonal field of view lens and a 1280×1024 pixel imaging sensor. Though the imaging sensor is 1280×1024, the video is acquired at the Common Intermediate Format (CIF) standard [39] size of 352×288 pixels at a rate of approximately 14.34 frames per second. The GPS receiver and compass-magnetometer measure the geoposition and relative orientation of the camera at 1 Hz and 4 Hz, respectively.

First, data was acquired for nearly 15 minutes while walking about a 40 m $\times$ 70 m building courtyard. The sensor platform was hand-held during the acquisition process with no special attention towards dampening movement associated with walking. Further, in order to emulate a helmet-mounted system, at times the sensor platform was oriented such that it followed the head pose of the person carrying it. Other times, the camera was generally pointed forward, in the direction of walking. In either case, the geoposition and orientation of the camera was reliably estimated using the approach described in section 1.3. Figure 1.9 illustrates the improved geopositioning precision of this approach over GPS-derived measurements alone.

For a quantitative analysis of precision, the total standard deviations of the GPS-derived geoposition measurements and Kalman filter a posteriori estimates are compared. The total standard deviation σ_{total} is defined as $\sigma_{\text{total}} = \sqrt{\text{trace}(\Sigma)}$, where Σ is the covariance matrix of the measurement or estimate of interest. Over the 11483 video frames acquired, the average total standard deviation of the GPS-derived measurements of geoposition is 57.735 m, while the average total standard deviation of the a posteriori state estimates for the same 11483 video frames is 37.501 m. For this data set, the addition of video and orientation sensors produced over a 35% increase in precision from using GPS alone.

Figure 1.10 shows the distance between the geoposition measurements and estimates. At times, there is over a 30 m difference between the two. For high-precision applications, such as georegistration of urban 3D scene reconstruction from ground-level video, 30 m may be critical.

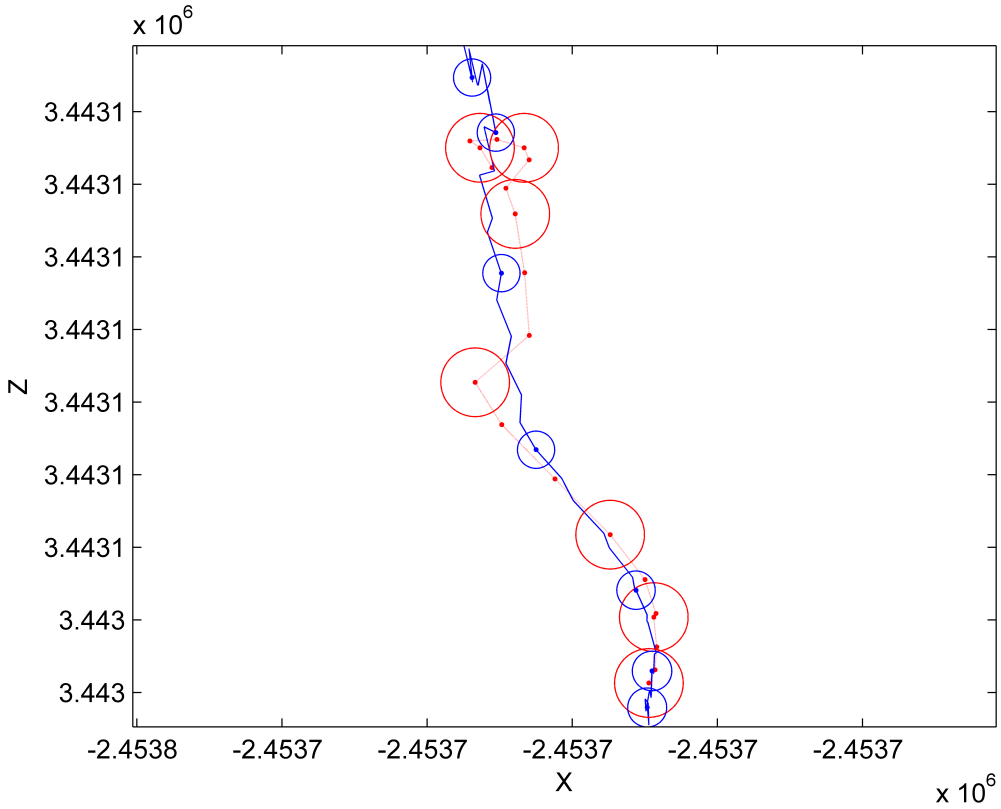


Figure 1.9: Geoposition estimates and their uncertainty. The estimated camera geoposition (in blue) at 1416 sequential time steps of a 11483 frame video. For comparison, the uncertainty ellipses of every third GPS-derived geoposition measurement (in red) and corresponding Kalman filter a posteriori estimates (in blue) are shown. The geopositions and their associated covariances are projected to the XZ-plane and 1% uncertainty bounds indicated. The uncertainty of the estimated geopositions is substantially smaller than the GPS-derived ones.

1.6 Conclusions

This dissertation has presented a new approach for precisely estimating the geoposition and orientation of one or more ground-level video cameras, where each calibrated camera is equipped with a GPS receiver and compass-magnetometer. Moreover, we address this problem using inexpensive consumer-grade sensors. The GPS receiver measures the latitude, longitude, and height above mean sea level of the video camera and the orientation of the video camera is derived from data acquired by a compass-magnetometer, which measures the pitch, roll, and yaw of the camera. Additionally, the camera motion between successive video frames is measured from features that have been

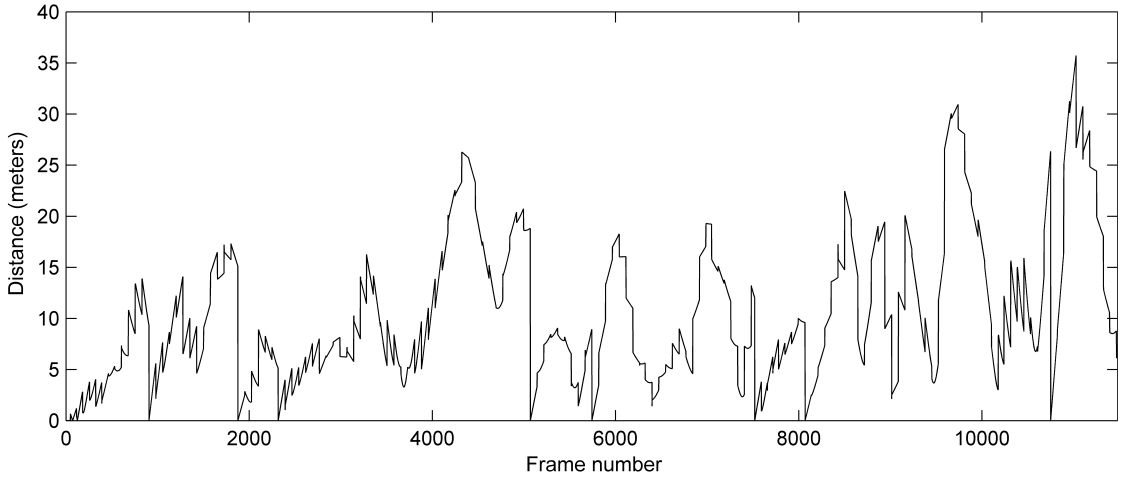


Figure 1.10: Distance between geoposition measurement and estimate. The distance between the GPS-derived geoposition measurement and the Kalman filter state estimate of geoposition over 11483 video frames. A distance of zero indicates that the state estimate of geoposition was reinitialized to the current measurement. Reinitialization is performed when the measurement is not within the 99.73% uncertainty bounds of the Kalman filter a priori estimate (prediction).

robustly tracked throughout the acquired video. We have described geospatial methods commonly used in photogrammetry for mapping measurements from this disparate set of sensors such that they are relative to the World Geodetic System 1984 geocentric coordinate frame, a world-wide standard Earth-centered, Earth-fixed Cartesian coordinate frame. The associated uncertainty of each measurement is mapped to this coordinate frame using a first-order nonlinear propagation of covariance model.

The geoposition and orientation of each camera is independently estimated from all of its associated sensor measurements using an extended Kalman filter. The filter uses the measurements provided by this combination of sensors to sequentially estimate the geoposition and orientation of the camera relative the WGS84 geocentric coordinate frame. The measurements and their associated uncertainties are input at different frequencies and are sometimes incomplete due to GPS dropouts, corrupt video frames, etc., yet it has been shown that the filter reliably estimates the most probable camera geoposition and orientation and their uncertainty at each video frame. Using this approach for combining GPS, 3-axis orientation, and video-derived measurements, experimental results indicate a 35% increase in the precision of the geoposition estimates over solely using GPS.

We have also described a method for further reducing the geoposition and orientation uncertainties in the case of multiple cameras imaging the same region of a scene. Our approach uses a combination of guided matching, feature descriptor comparison, and robust modeling fitting to determine the set of cross-camera feature correspondences between the images associated with all cameras imaging the same region of the scene. The resulting independent observations of corresponding features contained in the scene are used to jointly estimate the maximum likelihood of the geoposition and orientation of all cameras imaging the same region of a scene for which feature correspondences have been established. This approach yields decreased relative errors between the cameras, resulting in more precise estimates of the geoposition and orientation of the cameras.

This general approach allows the video cameras to be located anywhere in the proximity of the Earth. Further, this approach scales to multiple cameras at the cost of losing the cross covariance information between different cameras. It is expected that this work will enable other higher-level applications such as high-precision urban scene reconstruction from multi-camera video data. Although this work focuses on the precise estimation of geoposition and orientation, future work will include an evaluation of the accuracy of such estimates. Accuracy will be assessed by comparing the estimated geoposition to that of ground control points whose coordinates are very accurately known.

Acknowledgement This chapter, in full, is being prepared for publication in collaboration with S. Belongie. I am the primary investigator and author of this paper.

2

Sensors

Multiple sensors are required to precisely determine the geoposition and orientation of a video camera. Further, one objective of this dissertation is to address this problem using consumer-grade sensors that are inexpensive and produced in large volumes. This chapter describes the suite of sensors used to acquire data for the experiments contained in this dissertation and the preprocessing of the acquired data.

The sensor system includes a video camera, Global Positioning System (GPS) receiver, and 3-axis orientation sensor. Figure 1.1 on page 5 shows the configuration of these sensors. The GPS receiver measures the latitude, longitude, and height above mean sea level of the video camera. The orientation of the video camera is derived from data acquired by a compass-magnetometer, which measures the pitch, roll, and yaw of the sensors. Additional camera motion estimates can be made from the video data, but require that the camera that acquired the video is calibrated.

This chapter particularly details the mapping of the measurements from this set of disparate sensors to a common coordinate frame and the propagation of the uncertainty of the sensor measurements to this coordinate frame. In addition, camera calibration is discussed, including the model used to characterize the camera and the preprocessing calculations required for camera motion estimation from video.

2.1 Video camera

The data acquisition system contains a Microsoft LifeCam VX-6000 video camera [60]. The imaging components of this camera are a 72° diagonal field of view lens

and a 1280×1024 pixel imaging sensor. Though the imaging sensor is 1280×1024, the video is acquired at the Common Intermediate Format (CIF) standard [39] size of 352×288 pixels. The camera acquires video at a rate of approximately 14.34 frames per second.

The camera must be calibrated in order to estimate camera motion from video acquired by the camera. Camera calibration is the process of estimating the parameters of a model that characterize the projection of 3D world points to 2D image points under a given camera. The remainder of this section describes the camera model used and the calibration procedure.

2.1.1 Camera model

The image formation process maps 3D world points to 2D image points. Real camera lenses exhibit distortion that results in 2D image points that deviate from the images of the same 3D world points imaged under an ideal lens. Many camera models and calibration techniques have been developed to accurately reproduce this distortion (e.g., [100, 36, 94, 91, 11, 7, 27, 6, 12]). The model used in the work presented here is similar to [36]. In addition to modeling the set of internal camera parameters contained in the camera calibration matrix, described below, it models radial and tangential lens distortions. These types of distortions are generally the most significant and often capture other lens aberrations that are not explicitly modeled. Under this model, 3D world points are mapped to 2D image points as follows.

First, 3D world points $\widetilde{\mathbf{X}} = (\widetilde{X}, \widetilde{Y}, \widetilde{Z})^\top$ are transformed to the camera coordinate frame and projected to 2D normalized (homogeneous) coordinates $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{w})^\top$. This mapping is

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{w} \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \widetilde{X} \\ \widetilde{Y} \\ \widetilde{Z} \\ 1 \end{pmatrix}$$

$$\hat{\mathbf{x}} = [\mathtt{R} \mid \mathbf{t}] \begin{pmatrix} \widetilde{\mathbf{X}} \\ 1 \end{pmatrix}$$

$$\hat{\mathbf{x}} = \hat{\mathtt{P}}\mathbf{X}$$

where $\mathtt{R}$ and $\mathbf{t}$ are the 3D rotation and translation, respectively, that transform coor-

dinates in the world coordinate frame to coordinates in the camera coordinate frame. $\hat{\mathtt{P}} = [\mathtt{R} \mid \mathbf{t}]$ is called the normalized camera projection matrix and represents a camera with an ideal lens that maps 3D coordinates in the world coordinate frame to non-distorted normalized 2D coordinates. The relationship between 3D coordinates in the camera coordinate frame, image coordinates, and normalized coordinates is illustrated in figure 1.2 on page 6.

Short focal length or wide field of view lenses commonly exhibit distortions along radial directions. Radial distortion appears as either pincushion or barrel distortion of the image. Additionally, tangential distortion occurs when multiple or compound lenses are not aligned along their optical centers, a configuration referred to as lens decentering. In order to model the radial and tangential lens distortion present in a non-ideal lens, non-distorted normalized 2D coordinates $\hat{\tilde{\mathbf{x}}} = (\hat{\tilde{x}}, \hat{\tilde{y}})^\top = (\hat{x}/\hat{w}, \hat{y}/\hat{w})^\top$ are mapped to distorted normalized 2D coordinates $\hat{\tilde{\mathbf{x}}}_d = (\hat{\tilde{x}}_d, \hat{\tilde{y}}_d)^\top$ by

$$\begin{pmatrix} \hat{\tilde{x}} \\ \hat{\tilde{y}} \end{pmatrix} \mapsto \begin{pmatrix} \hat{\tilde{x}}_d \\ \hat{\tilde{y}}_d \end{pmatrix} = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6) \begin{pmatrix} \hat{\tilde{x}} \\ \hat{\tilde{y}} \end{pmatrix} + \begin{pmatrix} 2\kappa_3 \hat{\tilde{x}}\hat{\tilde{y}} + \kappa_4(r^2 + 2\hat{\tilde{x}}^2) \\ \kappa_3(r^2 + 2\hat{\tilde{y}}^2) + 2\kappa_4 \hat{\tilde{x}}\hat{\tilde{y}} \end{pmatrix} \tag{2.1}$$

where κ_1, κ_2, and κ_5 are radial distortion parameters, κ_3 and κ_4 are tangential distortion parameters, and $r = \sqrt{\hat{\tilde{x}}^2 + \hat{\tilde{y}}^2}$.

The final step of the imaging process is the mapping of distorted normalized coordinates $\hat{\tilde{\mathbf{x}}}_d = (\hat{\tilde{x}}_d, \hat{\tilde{y}}_d)^\top$ to distorted image coordinates $\tilde{\mathbf{x}}_d = (\tilde{x}_d, \tilde{y}_d)^\top$ by

$$\begin{aligned} \begin{pmatrix} \tilde{x}_d \\ \tilde{y}_d \\ 1 \end{pmatrix} &= \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \hat{\tilde{x}}_d \\ \hat{\tilde{y}}_d \\ 1 \end{pmatrix} \\ \begin{pmatrix} \tilde{\mathbf{x}}_d \\ 1 \end{pmatrix} &= \mathtt{K} \begin{pmatrix} \hat{\tilde{\mathbf{x}}}_d \\ 1 \end{pmatrix} \\ \mathbf{x}_d &= \mathtt{K}\hat{\mathbf{x}}_d \end{aligned} \tag{2.2}$$

where $(x_0, y_0)^\top$ is the principal point, s is the skew, and α_x and α_y are the focal lengths of the camera in the x and y directions. Two parameters for focal length are necessary to model imaging sensors with non-square pixels, which results in $\alpha_x \neq \alpha_y$. The matrix $\mathtt{K}$ is called the camera calibration matrix and encompasses the intrinsic parameters of the camera, less the distortion parameters. All of the parameters in $\mathtt{K}$ are in terms of pixel dimensions.

For conventional cameras, the above model adequately characterizes the image formation process. It is important to note that images acquired under such a camera model are in distorted image coordinates.

Inverse mapping

For applications needing to estimate camera rotation and translation (or position) from images, the distorted image coordinates must be mapped to non-distorted normalized coordinates—the inverse mapping process, $\tilde{\mathbf{x}}_d \mapsto \hat{\tilde{\mathbf{x}}}$, i.e., we must undistort $\hat{\mathbf{x}}_d = \mathtt{K}^{-1}\mathbf{x}_d$. Due to the high degree of (2.1), there is not a general algebraic expression for mapping distorted normalized coordinates to undistorted normalized coordinates, therefore the undistortion calculation must be performed numerically using one of many optimization methods.

The Levenberg-Marquardt optimization method [46, 55] is widely used in multiple view geometry [32] and is used throughout this dissertation. It is best described as a blend between the Gauss-Newton method and gradient descent [30, 70]. The Levenberg-Marquardt algorithm requires a measurement vector $\mathbf{X}$, its associated covariance matrix $\Sigma_{\mathbf{X}}$, an initial estimate of the parameter vector $\widehat{\mathbf{P}}$ being estimated, and the function that maps the current estimate of the parameter vector $\widehat{\mathbf{P}}$ to an estimate of the measurement vector $\widehat{\mathbf{X}}$. The algorithm iteratively finds the parameter vector $\widehat{\mathbf{P}}$ that minimizes $\boldsymbol{\epsilon}^\top \Sigma_{\mathbf{X}}^{-1}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = \mathbf{X} - \widehat{\mathbf{X}}$ is the error between the measurement and the estimated measurement. Central to Levenberg-Marquardt minimization is computation of the Jacobian matrix $\mathtt{J} = \partial\widehat{\mathbf{X}}/\partial\widehat{\mathbf{P}}$, which can be computed either numerically or using an analytical expression. For reasons of improved convergence and speed, an analytical expression is preferred. The covariance of the final estimate of the parameter vector is given by $\Sigma_{\widehat{\mathbf{P}}} = (\mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1}\mathtt{J})^{+}$.

Applied to the undistortion of normalized coordinates, $\hat{\tilde{\mathbf{x}}}_d \mapsto \hat{\tilde{\mathbf{x}}}$, the measurement vector is the distorted normalized coordinates $\hat{\tilde{\mathbf{x}}}_d$, the parameter vector to be estimated is the undistorted normalized coordinates $\hat{\tilde{\mathbf{x}}}$, the function that maps an estimate

of $\hat{\tilde{\mathbf{x}}}$ to $\hat{\tilde{\mathbf{x}}}_d$ is (2.1), and the analytical expression for the Jacobian is

$$\frac{\partial \hat{\tilde{\mathbf{x}}}_d}{\partial \hat{\tilde{\mathbf{x}}}} = 2(\kappa_1 + 2\kappa_2 r^2 + 3\kappa_5 r^4) \begin{bmatrix} \hat{\tilde{x}}^2 & \hat{\tilde{x}}\hat{\tilde{y}} \\ \hat{\tilde{x}}\hat{\tilde{y}} & \hat{\tilde{y}}^2 \end{bmatrix} + \begin{bmatrix} \gamma + 2(\kappa_3 \hat{\tilde{y}} + 2\kappa_4 \hat{\tilde{x}}(\hat{\tilde{x}}^2 + r^2)) & 2\hat{\tilde{x}}(\kappa_3 + 2\kappa_4 \hat{\tilde{x}}\hat{\tilde{y}} \\ 2\hat{\tilde{y}}(\kappa_4 + 2\kappa_3 \hat{\tilde{x}}\hat{\tilde{y}} & \gamma + 2(\kappa_4 \hat{\tilde{x}} + 2\kappa_3 \hat{\tilde{y}}(\hat{\tilde{y}}^2 + r^2)) \end{bmatrix}$$

where $\gamma = 1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6$. The initial estimate of $\hat{\tilde{\mathbf{x}}}$ is set to its distorted coordinates $\hat{\tilde{\mathbf{x}}}_d$.

The covariance matrix $\Sigma_{\hat{\tilde{\mathbf{x}}}_d}$ associated with the measurement vector $\hat{\tilde{\mathbf{x}}}_d$ is calculated from the covariances of $\mathtt{K}$ and $\mathbf{x}_d$ as they propagate through the equation $\hat{\mathbf{x}}_d = \mathtt{K}^{-1}\mathbf{x}_d$. For clarity, let $\mathtt{A} = \mathtt{K}^{-1}$, so

$$\begin{pmatrix} \hat{\tilde{\mathbf{x}}}_d \\ 1 \end{pmatrix} = \mathtt{A} \begin{pmatrix} \tilde{\mathbf{x}}_d \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \hat{\tilde{x}}_d \\ \hat{\tilde{y}}_d \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \tilde{x}_d \\ \tilde{y}_d \\ 1 \end{pmatrix}$$

Using this notation, $\Sigma_{\hat{\tilde{\mathbf{x}}}_d} \approx \mathtt{J}_{\mathbf{a}} \Sigma_{\mathbf{a}} \mathtt{J}_{\mathbf{a}}^\top + \mathtt{J}_{\tilde{\mathbf{x}}_d} \Sigma_{\tilde{\mathbf{x}}_d} \mathtt{J}_{\tilde{\mathbf{x}}_d}^\top$, where

$$\mathtt{J}_{\mathbf{a}} = \frac{\partial \hat{\tilde{\mathbf{x}}}_d}{\partial \mathbf{a}} = \begin{bmatrix} \tilde{x}_d & \tilde{y}_d & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{x}_d & \tilde{y}_d & 1 & 0 & 0 & 0 \end{bmatrix}, \; \mathtt{J}_{\tilde{\mathbf{x}}_d} = \frac{\partial \hat{\tilde{\mathbf{x}}}_d}{\partial \tilde{\mathbf{x}}_d} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}$$

and $\Sigma_{\mathbf{a}} \approx \mathtt{J}_{\mathbf{k}} \Sigma_{\mathbf{k}} \mathtt{J}_{\mathbf{k}}^\top$, where

$$\mathtt{J}_{\mathbf{k}} = \frac{\partial \mathbf{a}}{\partial \mathbf{k}} = \begin{bmatrix} -\frac{1}{\alpha_x^2} & 0 & 0 & \frac{s}{\alpha_y \alpha_x^2} & 0 & 0 & \frac{-\beta}{\alpha_y \alpha_x^2} & 0 & 0 \\ \frac{s}{\alpha_y \alpha_x^2} & -\frac{1}{\alpha_x \alpha_y} & 0 & -\frac{s^2}{\alpha_x^2 \alpha_y^2} & \frac{s}{\alpha_x \alpha_y^2} & 0 & \frac{s\beta}{\alpha_x^2 \alpha_y^2} & \frac{-\beta}{\alpha_x \alpha_y^2} & 0 \\ \frac{-\beta}{\alpha_y \alpha_x^2} & \frac{y_0}{\alpha_x \alpha_y} & -\frac{1}{\alpha_x} & \frac{s\beta}{\alpha_x^2 \alpha_y^2} & -\frac{sy_0}{\alpha_x \alpha_y^2} & \frac{s}{\alpha_x \alpha_y} & -\frac{\beta^2}{\alpha_x^2 \alpha_y^2} & \frac{y_0 \beta}{\alpha_x \alpha_y^2} & \frac{-\beta}{\alpha_x \alpha_y} \\ 0 & 0 & 0 & -\frac{1}{\alpha_x \alpha_y} & 0 & 0 & \frac{y_0}{\alpha_x \alpha_y} & 0 & 0 \\ 0 & 0 & 0 & \frac{s}{\alpha_x \alpha_y^2} & -\frac{1}{\alpha_y^2} & 0 & -\frac{sy_0}{\alpha_x \alpha_y^2} & \frac{y_0}{\alpha_y^2} & 0 \\ 0 & 0 & 0 & \frac{-\beta}{\alpha_x \alpha_y^2} & \frac{y_0}{\alpha_y^2} & -\frac{1}{\alpha_y} & \frac{y_0 \beta}{\alpha_x \alpha_y^2} & -\frac{y_0^2}{\alpha_y^2} & \frac{y_0}{\alpha_y} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\alpha_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{s}{\alpha_x \alpha_y} & -\frac{1}{\alpha_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-\beta}{\alpha_x \alpha_y} & \frac{y_0}{\alpha_y} & -1 \end{bmatrix}$$

where $\beta = sy_0 - x_0 \alpha_y$.

The covariance matrix $\Sigma_{\tilde{\mathbf{x}}_d}$ is a function of how the distorted image coordinates $\tilde{\mathbf{x}}_d$ are measured, whether manually or automatically (e.g., by an autonomous feature detector). If the covariance of the measured coordinates is unknown, it is assumed that $\Sigma_{\tilde{\mathbf{x}}_d}$ is the identity matrix. The covariance matrix $\Sigma_{\mathbf{k}}$ is determined during the camera calibration process that is described next.

2.1.2 Camera calibration

Camera calibration is the process of estimating the internal parameters of a given camera model. In this work, the internal parameters of the camera model described in the previous section are estimated using [4], which also calculates the uncertainty of the estimated parameters contained in $\mathtt{K}$ (see (2.2)). Calibration is performed from multiple images of a known calibration target, in this case a planar black and white checkerboard pattern, 7 squares $\times$ 9 squares where each square is 30 mm $\times$ 30 mm. 41 images of the calibration target were acquired with the target at varying orientations and distances from the camera. Sample images of the calibration target are shown in figure 2.1.

For each image of the calibration target, the image coordinates of each corner of the squares must be determined. This is accomplished in an assisted manner. First, a human manually selects the image of the four corners that are closest to the bounding corners of the entire target. From this, the calibration application automatically predicts the locations of the remaining image corners. If the predicted corners are acceptable to the human, then the coordinates of the corners are determined to subpixel accuracy. Otherwise, the human may either make an initial guess of the distortion values to improve the prediction or manually guide the predictions, which are then determined to subpixel accuracy. This same procedure is done for each image of the calibration target.

The parameters of the camera model are then estimated from the sets of image corners and their corresponding coordinates on the planar calibration target. This is completed by a parameter initialization phase, followed by a nonlinear optimization procedure that minimizes the error between the measured image coordinates and the projection of the target coordinates. The initialization process computes a closed-form solution for all camera parameters except the distortion parameters. All of the camera parameters, including those for distortion, are then estimated using the gradient descent optimization method, where the Jacobian matrix is calculated from an analytical expression. The calibration results are found in table 2.1. The covariance matrix $\Sigma_{\mathbf{k}}$ associated with

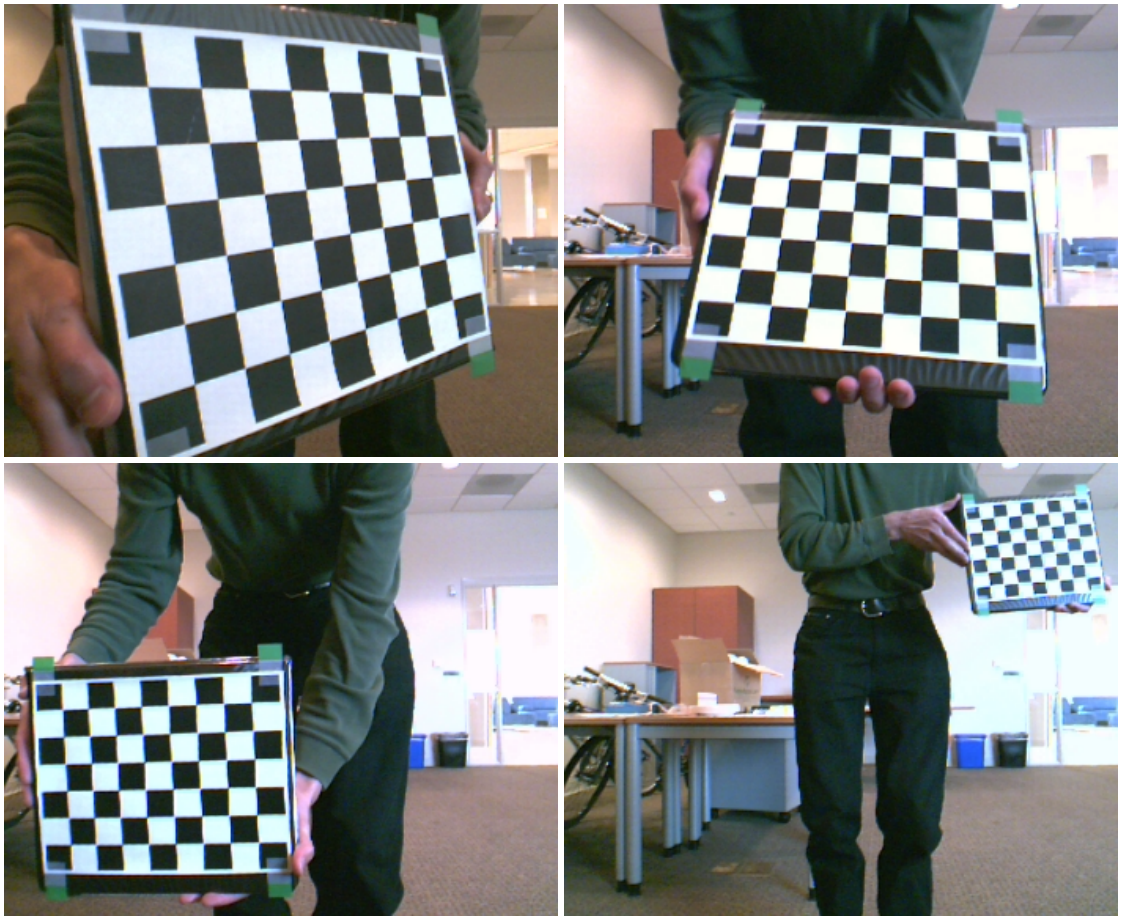


Figure 2.1: Images of the camera calibration target. 4 of the 41 images of the calibration target used to calibrate the camera.

the camera calibration matrix $\mathtt{K}$ is given by $\Sigma_{\mathbf{k}} = \mathrm{diag}(\sigma^2_{\alpha_x}, \sigma^2_s, \sigma^2_{x_0}, 0, \sigma^2_{\alpha_y}, \sigma^2_{y_0}, 0, 0, 0)$.

2.2 GPS receiver

The data acquisition system measures the latitude, longitude, and height above mean sea level of the camera using a Holux GPSlim236 GPS receiver [37]. The GPS receiver measures the height above mean sea level and geodetic latitude and longitude at a frequency of 1 Hz. For use in subsequent computations, these quantities are transformed to an Earth-centered, Earth-fixed Cartesian coordinate system as described in the following section.

Table 2.1: Estimated internal camera parameters and their uncertainty

	Estimate	Standard Deviation σ
α_x	354.263	0.382
s	0.000	0.000
x_0	177.577	0.192
α_y	354.354	0.367
y_0	134.144	0.156
κ_1	−0.075	
κ_2	0.000	
κ_3	0.000	
κ_4	−0.001	
κ_5	0.000	

2.2.1 Geodetic coordinate transformation

Geodetic coordinates and their transformation have been extensively studied in the field of geospatial science [78, 79, 97]. Definitions of the following mapping, charting, and geodesy [19] terms will facilitate further discussion.

geoid The equipotential surface in the gravity field of the Earth which approximates the undisturbed mean sea level extended continuously through the continents.

reference ellipsoid A theoretical figure whose dimensions closely approach the dimensions of the geoid. The exact dimensions of the ellipsoid are determined by various considerations of the section of the Earth's surface concerned.

geoid-ellipsoid separation The distance between the geoid and the mathematical reference ellipsoid as measured along the ellipsoidal normal.

datum The reference frame that measurements are made relative to. The datum is expressed as the parameters of the reference ellipsoid used by the reference frame, and the origin and orientation of the coordinate system of the datum.

geodetic coordinates The quantities of latitude, longitude, and (ellipsoid) height, which define the position of a point on the surface of the Earth with respect to the reference ellipsoid.

geocentric coordinates Coordinates that define the position of a point with respect to the center of the Earth.

Over 225 datums are commonly used in mapping, charting, and geodesy. Each of these datums is associated with one of the 23 reference ellipsoids shown in table 2.2. Unique among these is the World Geodetic System (WGS) [20] because it is both a datum and a reference ellipsoid. As such, WGS provides the means for relating positions on various datums to an Earth-centered, Earth-fixed coordinate system.

The Earth Gravity Model 1996 (EGM96) [45] is a geopotential model of the Earth consisting of spherical harmonic coefficients complete to degree and order 360. This geopotential model is used as a geodetic reference to convert between EGM96 geoid height (i.e., height above mean sea level) to height above WGS84 ellipsoid, thereby correcting for any geoid-ellipsoid separation.

Independent of both datum and reference ellipsoid is the reference frame of the coordinates. There are 33 common reference frames comprised of different coordinate systems, map projections, grids, and grid reference systems [18, 17, 89]. The general transformation of coordinates is performed using the approach described in [97], which is summarized as:

1. Convert the input coordinates from the input reference frame to the geodetic reference frame.
2. Shift the intermediate geodetic coordinates from the input datum to WGS84.
3. Convert from EGM96 geoid height to WGS84 ellipsoid height, if needed.
4. Shift the shifted intermediate WGS84 geodetic coordinates to the output datum.
5. Convert the shifted intermediate geodetic coordinates to the output coordinate reference frame.

For use in subsequent computations, the GPS receiver measurements of latitude, longitude, and height are transformed to geocentric coordinates. The relationship between the geodetic and geocentric coordinate systems is shown in figure 1.3 on page 9. For GPS, the current underlying coordinate system is WGS84—GPS receiver measurements are in WGS84 geodetic coordinates with EGM96 geoid height. For the work presented in this dissertation, WGS84 geocentric is used as the principal coordinate

Table 2.2: Common reference ellipsoids and their parameters

Reference Ellipsoid		Semi-major Axis (m)	Semi-minor Axis (m)
Airy 1830		6377563.396	6356256.9090
Modified Airy		6377340.189	6356034.4480
Australian National		6378160.000	6356774.7190
Bessel 1841	Namibia	6377483.865	6356165.3830
	Ethiopia, Indonesia, Japan, Korea	6377397.155	6356078.9630
Clarke 1866		6378206.400	6356583.8000
Clarke 1880		6378249.145	6356514.8700
Everest	India 1830	6377276.345	6356075.4130
	E. Malaysia & Brunei	6377298.556	6356097.5500
	India 1956	6377301.243	6356100.2280
	W. Malaysia 1969	6377295.664	6356094.6680
	W. Malaysia & Singapore 1948	6377304.063	6356103.0390
	Pakistan	6377309.613	6356109.5710
Modified Fischer 1960 (South Asia)		6378155.000	6356773.3200
Helmert 1906		6378200.000	6356818.1700
Hough 1960		6378270.000	6356794.3430
Indonesian 1974		6378160.000	6356774.5040
International 1924		6378388.000	6356911.9460
Krassovsky 1940		6378245.000	6356863.0190
Geodetic Reference System 1980 (GRS80)		6378137.000	6356752.3141
South American 1969		6378160.000	6356774.7190
World Geodetic System 1972 (WGS72)		6378135.000	6356750.5200
World Geodetic System 1984 (WGS84)		6378137.000	6356752.3142

frame, as doing so minimizes geodetic coordinate transformations. To transform GPS receiver measurements to WGS84 geocentric coordinates, the above procedure simplifies to:

1. Convert from EGM96 geoid height to WGS84 ellipsoid height.
2. Convert the height-corrected geodetic coordinates to geocentric coordinates.

After coordinate conversion, the resulting coordinates reside in an Earth-centered, Earth-fixed Cartesian coordinate system, as desired.

2.2.2 GPS positioning uncertainty

The Global Positioning System (GPS) [64, 75, 76] is the most accurate world-wide navigation system developed to date. GPS was developed by the United States Department of Defense and presently consists of more than two dozen satellites, each with a highly accurate atomic clock, that orbit the Earth. Each satellite periodically transmits signals that report the satellite position and the transmission time. GPS receivers use these satellite messages to calculate the range to three or more satellites and then determine the position of the receiver using trilateration. However, GPS-derived positioning is not without error [56, 13, 5, 57, 74]. This section gives a brief description of the several sources of error that contribute to its positioning error.

A GPS receiver fundamentally estimates the range to a given satellite, corrupted by a user clock bias. This quantity is called the pseudorange ρ and is calculated as

$$\rho = c(t_{a_u} - t_{t_s}) \tag{2.3}$$

where c is the speed of light in a vacuum, t_{a_u} is the arrival time measured by the user, and t_{t_s} is the value of the transmission time in the current satellite message. The true range

$$D = \|\mathbf{P}_s - \mathbf{P}_u\| \tag{2.4}$$

is the distance between the true satellite position $\mathbf{P}_s$ and the true user position $\mathbf{P}_u$. Any error between the pseudorange and true range or between the transmitted satellite position and the true satellite position results in receiver positioning error.

The first potential source of error is the satellite clock. If such error exists, the satellite transmit time t_{t_s} will be in error. This error is modeled as

$$t_{t_s} = t_t + b_s \tag{2.5}$$

where t_t is the true transmit time and b_s is the true error in the transmission time of the satellite. The arrival time measured by the user t_{a_u} characterizes all remaining errors, including clock bias of the user and other measurement errors, and is given by

$$\begin{aligned} t_{a_u} &= t_t + \|\mathbf{P}_s - \mathbf{P}_u\|/c + T + I + b_u + v \\ t_{a_u} &= t_a + b_u + v \end{aligned} \tag{2.6}$$

where T is the true tropospheric delay, I is the true ionospheric delay, b_u is the user clock bias estimate common to a set of simultaneous measurements, and v is the receiver noise, multipath error, and interchannel error. The true signal arrival time $t_a = t_t + D/c + T + I$ is modeled as the true signal transmission time delayed by the vacuum transit time and additional true delays caused by the ionosphere and troposphere.

Substituting (2.4), (2.5), and (2.6) into (2.3) yields

$$\rho = \|\mathbf{P}_s - \mathbf{P}_u\| + c(b_u - b_s + T + I + v)$$

Errors in the variables of this equation fall into one of the following standard classes of GPS errors: ephemeris (errors in the transmitted satellite position), satellite clock (errors in the satellite transmit time), ionosphere (errors due to ionospheric effects), troposphere (errors due to tropospheric effects), multipath (errors caused by multiple, reflected signals entering the receiver antenna), and receiver (errors due to thermal noise, software accuracy, and interchannel biases inherent in the receiver). The magnitude of each of these error sources is summarized in the standard GPS error shown in table 1.1 on page 11. The magnitude of the error of the position along the ellipsoidal normal σ_{vertical} and in the plane orthogonal to the ellipsoidal normal $\sigma_{\text{horizontal}}$ are derived from the standard deviations of the standard error sources. The standard GPS errors assume the median geometric configuration of the satellites. To account for differing satellite geometry, the work presented in this dissertation uses greater values than those in the standard error table. Specifically, the covariance of the coordinates of the camera center in the WGS84 geocentric coordinate frame $\Sigma_{\widetilde{\mathbf{C}}} = \Sigma_{\mathbf{P}_u} = \text{diag}\left(\sigma^2_{\text{GPS}}, \sigma^2_{\text{GPS}}, \sigma^2_{\text{GPS}}\right)$ where $\sigma_{\text{GPS}} = 33.3\,\text{meters}$.

2.3 3-axis orientation sensor

The Advanced Orientation Systems EZ-COMPASS-3 [2], a tilt compensated compass-magnetometer, is used to measure 3-axis orientation. This sensor characterizes

3D orientation by Euler angles in the so-called "XYZ" convention. In this convention, the rotation is given by pitch θ, roll ψ, and yaw ϕ angles. The angles define a rotated coordinate frame relative to a local (unrotated) coordinate frame with origin at the current position of the sensor, positive X-axis pointing north, positive Y-axis pointing west, and positive Z-axis pointing up, along the ellipsoidal normal. This is shown in figure 1.4 on page 12.

The 3D orientation of the sensor is calculated as the composition of three rotations, a first rotation by an angle ϕ about the Z-axis, a second by an angle θ about the Y-axis, and a third by an angle ψ about the X-axis. The 3D rotation matrix $\mathtt{R}$ that maps coordinates in the rotated coordinate frame to coordinates in the unrotated coordinate frame is formed from the pitch, roll, and yaw measurements by

$$\mathtt{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathtt{R} = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ \sin\psi\sin\theta\cos\phi - \cos\psi\sin\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\theta & \cos\theta\sin\psi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\theta & \cos\theta\cos\psi \end{bmatrix} \tag{2.7}$$

Using covariance propagation, the covariance matrix associated with the rotation matrix is given by $\Sigma_{\mathbf{r}} \approx \mathtt{J}_{\theta,\psi,\phi}\Sigma_{\theta,\psi,\phi}\mathtt{J}_{\theta,\psi,\phi}^{\top}$, where $\Sigma_{\theta,\psi,\phi} = \mathrm{diag}(\sigma_\theta^2, \sigma_\psi^2, \sigma_\phi^2)$ and

$$\mathtt{J}_{\theta,\psi,\phi} = \frac{\partial \mathbf{r}}{\partial(\theta,\psi,\phi)} = \begin{bmatrix} -ab & 0 & -cd \\ -ad & 0 & cb \\ -c & 0 & 0 \\ ecb & fab + ed & -ead - fb \\ ecd & fad - eb & eab - fd \\ -ea & fc & 0 \\ fcb & -eab + fd & -fad + eb \\ fcd & -ead - fb & fab + ed \\ -fa & -ec & 0 \end{bmatrix} \tag{2.8}$$

where $a = \sin\theta$, $c = \cos\theta$, $e = \sin\psi$, $f = \cos\psi$, $d = \sin\phi$, and $b = \cos\phi$. Table 2.3 shows the standard deviation of the pitch, roll, and yaw sensor measurements [54].

Table 2.3: Tilt compensated compass-magnetometer sensor error

Quantity	Standard Deviation σ (degrees)
Pitch θ	0.089
Roll ψ	0.089
Yaw ϕ	0.178

2.3.1 Rotation to camera coordinate frame and its uncertainty

As described above, the pitch, roll, and yaw measurements are relative to a local unrotated coordinate frame. Subsequent processing requires knowledge of the rotation that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame. This section describes the calculation of this rotation from GPS receiver measurements of the coordinates of the camera center in the WGS84 geocentric coordinate frame $\widetilde{\mathbf{C}}$ and its associated covariance $\Sigma_{\widetilde{\mathbf{C}}}$, and 3-axis orientation measurements of pitch θ, roll ψ, and yaw ϕ and the associated covariance $\Sigma_{\theta,\psi,\phi}$.

The rotation that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame is a composition of four rotations that transform coordinates through the following coordinate frames:

(a) WGS84 geocentric coordinate frame

(b) WGS84 local Cartesian coordinate frame

(c) Pitch-roll-yaw unrotated local Cartesian coordinate frame

(d) Pitch-roll-yaw rotated local Cartesian coordinate frame

(e) Camera coordinate frame

Of these, WGS84 local Cartesian has not yet been described. The WGS84 local Cartesian coordinate frame is similar to the pitch-roll-yaw unrotated local Cartesian coordinate frame shown in figure 1.4 on page 12. The difference for WGS84 local Cartesian is that the positive X-axis points east and positive Y-axis points north. As with the pitch-roll-yaw unrotated local Cartesian coordinate frame, the Z-axis points up. The relationship between these 5 coordinate frames is show in figure 1.5 on page 14.

Most of the rotation matrices that transform coordinates from a given coordinate frame to the camera coordinate frame are straightforward to calculate. The rotation

$\mathtt{R}_{d,e}$ from the pitch-roll-yaw rotated local Cartesian coordinate frame to the camera coordinate frame is given by

$$\mathtt{R}_{d,e} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

The rotation $\mathtt{R}_{c,d}$ from the pitch-roll-yaw unrotated local Cartesian coordinate frame to the pitch-roll-yaw rotated local Cartesian coordinate frame is calculated as $\mathtt{R}_{c,d} = \mathtt{R}_{d,c}^{\top}$, where $\mathtt{R}_{d,c}$ is given by (2.7) with Jacobian matrices $\partial \mathbf{r}_{d,c}/\partial(\theta, \psi, \phi)$ and $\partial \mathbf{r}_{c,d}/\partial \mathbf{r}_{d,c}$ given by (2.8) and (A.3), respectively.

The rotation $\mathtt{R}_{c,e}$ from the pitch-roll-yaw unrotated local Cartesian coordinate frame to the camera coordinate frame is the composition of the rotation $\mathtt{R}_{c,d}$ from the pitch-roll-yaw unrotated local Cartesian coordinate frame to the pitch-roll-yaw rotated local Cartesian coordinate frame and the rotation $\mathtt{R}_{d,e}$ from the pitch-roll-yaw rotated local Cartesian coordinate frame to the camera coordinate frame. This is calculated as $\mathtt{R}_{c,e} = \mathtt{R}_{d,e}\mathtt{R}_{c,d}$ with Jacobian matrices $\partial \mathbf{r}_{c,e}/\partial \mathbf{r}_{d,e}$ and $\partial \mathbf{r}_{c,e}/\partial \mathbf{r}_{c,d}$ given by (A.12) and (A.13), respectively.

The rotation $\mathtt{R}_{b,c}$ from the WGS84 local Cartesian coordinate frame to the pitch-roll-yaw unrotated local Cartesian coordinate frame is given by

$$\mathtt{R}_{b,c} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Similar to the composition of rotations above, the rotation $\mathtt{R}_{b,e}$ from the WGS84 local Cartesian coordinate frame to the camera coordinate frame is given by $\mathtt{R}_{b,e} = \mathtt{R}_{c,e}\mathtt{R}_{b,c}$ with Jacobian matrices $\partial \mathbf{r}_{b,e}/\partial \mathbf{r}_{c,e}$ and $\partial \mathbf{r}_{b,e}/\partial \mathbf{r}_{b,c}$ given by (A.12) and (A.13), respectively.

The remaining rotation $\mathtt{R}_{a,b}$ from the WGS84 geocentric coordinate frame to the WGS84 local Cartesian coordinate frame is more involved, as it first requires establishment of the WGS84 local Cartesian coordinate frame. The origin of the WGS84 local Cartesian coordinate frame is at the camera center $\widetilde{\mathbf{C}}$, converted to WGS84 geodetic coordinates $(\phi, \lambda, h)^{\top}$. Next, the origin and each of the three standard basis vectors in the

WGS84 geocentric coordinate frame are converted to WGS84 local Cartesian coordinates

$$(0,0,0)^\top \mapsto \mathbf{X}_0^\top$$
$$(1,0,0)^\top \mapsto \mathbf{X}_1^\top$$
$$(0,1,0)^\top \mapsto \mathbf{X}_2^\top$$
$$(0,0,1)^\top \mapsto \mathbf{X}_3^\top$$

The rotation $\mathtt{R}_{a,b}$ from the WGS84 geocentric coordinate frame to the WGS84 local Cartesian coordinate frame is then calculated as $\mathtt{R}_{a,b} = [\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_2 - \mathbf{X}_0 \mid \mathbf{X}_3 - \mathbf{X}_0]$. For the coordinate transformations described in this paragraph, the 5-step procedure described on page 41 is used. The Jacobian matrices $\partial(\phi, \lambda, h)/\partial\widetilde{\mathbf{C}}$ and $\partial\mathbf{r}_{a,b}/\partial(\phi, \lambda, h)$ are computed by numerical differentiation. Finally, the rotation $\mathtt{R}_{a,e}$ from the WGS84 geocentric coordinate frame to the camera coordinate frame is given by $\mathtt{R}_{a,e} = \mathtt{R}_{b,e}\mathtt{R}_{a,b}$, again with Jacobian matrices $\partial\mathbf{r}_{a,e}/\partial\mathbf{r}_{b,e}$ and $\partial\mathbf{r}_{a,e}/\partial\mathbf{r}_{a,b}$ analytically calculated using (A.12) and (A.13), respectively. For a minimal parameterization of a 3D rotation, $\mathtt{R}_{a,e}$ is mapped to exponential coordinates $\boldsymbol{\omega}$ with Jacobian $\partial\boldsymbol{\omega}/\partial\mathbf{r}_{a,e}$ as described in the following section.

Exponential coordinates for rotations

Euler's rotation theorem states that an arbitrary rotation in three dimensions may be described by only three parameters (e.g., Euler angles, see page 44). A rotation matrix $\mathtt{R} \in SO(3)$ uses the nine entries of $\mathtt{R}$ to characterize a 3D rotation and is therefore an overparameterization of the rotation. An alternative minimal parameterization that is commonly used in robotics and computer vision is exponential coordinates [66, 32, 53]. In [32], exponential coordinates are called the angle-axis representation of a rotation, which better describes this minimal parameterization of a rotation. Using exponential coordinates, a 3D rotation is parameterized by the 3-vector $\boldsymbol{\omega}$ that represents a rotation by an angle $\theta = \|\boldsymbol{\omega}\|$ about the axis $\boldsymbol{\omega}$. Note that a given rotation is not uniquely described by exponential coordinates since $2\pi n\boldsymbol{\omega}$ is the same rotation for all values of n in the set of positive integers $\mathbb{Z}^+$. To avoid the singularity at $\|\boldsymbol{\omega}\| = 2\pi$, it is good practice to ensure that $\|\boldsymbol{\omega}\| \leq \pi$. The check is simple; if $\|\boldsymbol{\omega}\| > \pi$, then replace $\boldsymbol{\omega}$ with $\boldsymbol{\omega}(1 - 2\pi/\|\boldsymbol{\omega}\|)$, which is the equivalent rotation.

Of specific interest is the mapping from exponential coordinates $\boldsymbol{\omega}$ to a rotation matrix $\mathtt{R}$ that represents the same rotation. This mapping is the so-called exponential

map $\mathtt{R} = \exp(\boldsymbol{\omega})$ and its inverse is simply called the inverse exponential map $\boldsymbol{\omega} = \log(\mathtt{R})$. The mathematics of these mappings is described as follows.

The exponential map is calculated by

$$\mathtt{R} = \exp(\boldsymbol{\omega}) = \cos\|\boldsymbol{\omega}\|\mathtt{I} + \frac{\sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|}[\boldsymbol{\omega}]_\times + \frac{1-\cos\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2}\boldsymbol{\omega}\boldsymbol{\omega}^\top$$

where $[\boldsymbol{\omega}]_\times$ is the skew-symmetric matrix corresponding to $\boldsymbol{\omega}$ (see (A.14)). The Jacobian matrix $\partial\mathbf{r}/\partial\boldsymbol{\omega}$ is given by

$$\begin{aligned}\frac{\partial\mathbf{r}}{\partial\boldsymbol{\omega}} = & -\operatorname{vec}(\mathtt{I})\sin\|\boldsymbol{\omega}\|\frac{\mathrm{d}\|\boldsymbol{\omega}\|}{\partial\boldsymbol{\omega}} + \frac{\sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|}\frac{\partial\operatorname{vec}(([\boldsymbol{\omega}]_\times)^\top)}{\partial\boldsymbol{\omega}} \\ & + \operatorname{vec}(([\boldsymbol{\omega}]_\times)^\top)\frac{\|\boldsymbol{\omega}\|\cos\|\boldsymbol{\omega}\| - \sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2}\frac{\mathrm{d}\|\boldsymbol{\omega}\|}{\partial\boldsymbol{\omega}} + s\frac{\partial\mathbf{m}}{\partial\boldsymbol{\omega}} + \mathbf{m}\frac{\mathrm{d}s}{\partial\boldsymbol{\omega}}\end{aligned} \tag{2.9}$$

where $\mathrm{d}\|\boldsymbol{\omega}\|/\partial\boldsymbol{\omega}$ and $\partial\operatorname{vec}(([\boldsymbol{\omega}]_\times)^\top)/\partial\boldsymbol{\omega}$ are given by (A.1) and (A.14), respectively, and $s = (1-\cos\|\boldsymbol{\omega}\|)/\|\boldsymbol{\omega}\|^2$ and $\mathtt{M} = \boldsymbol{\omega}\boldsymbol{\omega}^\top$ with Jacobian matrices $\partial\mathbf{m}/\partial\boldsymbol{\omega} = \boldsymbol{\omega}\otimes\mathtt{I} + \mathtt{I}\otimes\boldsymbol{\omega}$ and

$$\frac{\mathrm{d}s}{\partial\boldsymbol{\omega}} = \frac{\|\boldsymbol{\omega}\|\sin\|\boldsymbol{\omega}\| - 2(1-\cos\|\boldsymbol{\omega}\|)}{\|\boldsymbol{\omega}\|^3}\frac{\mathrm{d}\|\boldsymbol{\omega}\|}{\partial\boldsymbol{\omega}}$$

The inverse exponential map is not as straightforward. First the rotation axis $\mathbf{v}$ is calculated by solving $(\mathtt{R}-\mathtt{I})\mathbf{v} = \mathtt{A}\mathbf{v} = \mathbf{0}$, i.e., $\mathbf{v}$ is the null space of $\mathtt{A} = \mathtt{R}-\mathtt{I}$. The Jacobian $\partial\mathbf{a}/\partial\mathbf{r}$ is given by (A.10) and $\partial\mathbf{v}/\partial\mathbf{a}$ is calculated using the analytical method of [73]. The rotation angle θ is calculated by

$$\theta = \tan^{-1}\left(\frac{\sin\theta}{\cos\theta}\right)$$

where $\cos\theta = (\operatorname{trace}(\mathtt{R})-1)/2$ and $\sin\theta = (\mathbf{v}^\top\hat{\mathbf{v}})/2$, where $\hat{\mathbf{v}} = (r_{32}-r_{23}, r_{13}-r_{31}, r_{21}-r_{12})^\top$. And finally

$$\boldsymbol{\omega} = \theta\bar{\mathbf{v}}$$

where $\bar{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$ and the Jacobian $\partial\boldsymbol{\omega}/\partial\mathbf{r}$ is given by

$$\frac{\partial\boldsymbol{\omega}}{\partial\mathbf{r}} = \frac{\partial\boldsymbol{\omega}}{\partial\bar{\mathbf{v}}}\frac{\partial\bar{\mathbf{v}}}{\partial\mathbf{r}} + \frac{\partial\boldsymbol{\omega}}{\mathrm{d}\theta}\frac{\mathrm{d}\theta}{\partial\mathbf{r}} \tag{2.10}$$

where

$$\frac{\partial \hat{\mathbf{v}}}{\partial \mathbf{r}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\mathrm{d}\sin\theta}{\partial \mathbf{v}} = \frac{1}{2}\hat{\mathbf{v}}^\top \qquad \frac{\mathrm{d}\sin\theta}{\partial \hat{\mathbf{v}}} = \frac{1}{2}\mathbf{v}^\top$$

$$\frac{\mathrm{d}\sin\theta}{\partial \mathbf{r}} = \frac{\mathrm{d}\sin\theta}{\mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{r}} + \frac{\mathrm{d}\sin\theta}{\partial \hat{\mathbf{v}}}\frac{\partial \hat{\mathbf{v}}}{\partial \mathbf{r}}$$

$$\frac{\mathrm{d}\cos\theta}{\mathrm{d}\,\mathrm{trace}(\mathtt{R})} = \frac{1}{2} \qquad \frac{\mathrm{d}\theta}{\mathrm{d}\sin\theta} = \frac{\cos\theta}{\sin^2\theta + \cos^2\theta} \qquad \frac{\mathrm{d}\theta}{\mathrm{d}\cos\theta} = \frac{-\sin\theta}{\sin^2\theta + \cos^2\theta}$$

$$\frac{\partial \theta}{\partial \mathbf{r}} = \frac{\mathrm{d}\theta}{\mathrm{d}\sin\theta}\frac{\mathrm{d}\sin\theta}{\partial \mathbf{r}} + \frac{\mathrm{d}\theta}{\mathrm{d}\cos\theta}\frac{\mathrm{d}\cos\theta}{\mathrm{d}\,\mathrm{trace}(\mathtt{R})}\frac{\mathrm{d}\,\mathrm{trace}\,(\mathtt{R})}{\partial \mathbf{r}}$$

$$\frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{v}} = \frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{v}} + \frac{\partial \bar{\mathbf{v}}}{\mathrm{d}\|\mathbf{v}\|}\frac{\mathrm{d}\|\mathbf{v}\|}{\partial \mathbf{v}}$$

$$\frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{r}} = \frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{r}}$$

and $\mathrm{d}\,\mathrm{trace}(\mathtt{R})/\partial \mathbf{r}$, $\mathrm{d}\|\mathbf{v}\|/\partial \mathbf{v}$, $\partial \bar{\mathbf{v}}/\partial \mathbf{v}$, $\partial \bar{\mathbf{v}}/\mathrm{d}\|\mathbf{v}\|$, $\partial \boldsymbol{\omega}/\mathrm{d}\theta$, and $\partial \boldsymbol{\omega}/\partial \bar{\mathbf{v}}$ are given by equations found in section A.1 on page 78.

Uncertainty of rotation to camera coordinate frame

As detailed in this section, the rotation that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame $\boldsymbol{\omega}$ is dependent on the GPS receiver measurements of the coordinates of the camera center in the WGS84 geocentric coordinate frame $\widetilde{\mathbf{C}}$ and the 3-axis orientation measurements of pitch θ, roll ψ, and yaw ϕ. As such, in order to correctly model the uncertainty of $\boldsymbol{\omega}$, its joint covariance with $\widetilde{\mathbf{C}}$ must be calculated as follows.

$$\Sigma_{(\boldsymbol{\omega}^\top, \widetilde{\mathbf{C}}^\top)} \approx \mathtt{J}_{(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} \begin{bmatrix} \Sigma_{(\theta,\psi,\phi)} & \mathtt{0} \\ \mathtt{0} & \Sigma_{\widetilde{\mathbf{C}}} \end{bmatrix} \mathtt{J}^\top_{(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)}$$

where

$$\mathtt{J}_{(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} = \frac{\partial(\boldsymbol{\omega}^\top, \widetilde{\mathbf{C}}^\top)}{\partial(\theta,\psi,\phi,\widetilde{\mathbf{C}}^\top)} = \begin{bmatrix} \frac{\partial \boldsymbol{\omega}}{\partial(\theta,\psi,\phi)} & \frac{\partial \boldsymbol{\omega}}{\partial \widetilde{\mathbf{C}}} \\ \mathtt{0} & \mathtt{I} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial(\theta,\psi,\phi)} = \frac{\partial \boldsymbol{\omega}}{\partial \mathbf{r}_{a,e}}\frac{\partial \mathbf{r}_{a,e}}{\partial \mathbf{r}_{b,e}}\frac{\partial \mathbf{r}_{b,e}}{\partial \mathbf{r}_{c,e}}\frac{\partial \mathbf{r}_{c,e}}{\partial \mathbf{r}_{c,d}}\frac{\partial \mathbf{r}_{c,d}}{\partial \mathbf{r}_{d,c}}\frac{\partial \mathbf{r}_{d,c}}{\partial(\theta,\psi,\phi)}$$

$$\text{and } \frac{\partial \boldsymbol{\omega}}{\partial \widetilde{\mathbf{C}}} = \frac{\partial \boldsymbol{\omega}}{\partial \mathbf{r}_{a,e}}\frac{\partial \mathbf{r}_{a,e}}{\partial \mathbf{r}_{a,b}}\frac{\partial \mathbf{r}_{a,b}}{\partial(\phi,\lambda,h)}\frac{\partial(\phi,\lambda,h)}{\partial \widetilde{\mathbf{C}}}$$

3

Single Camera Estimation

For each video camera, initial measurements from the GPS receiver and 3-axis sensor are used to derive measurements of the geoposition and orientation of the camera. Additionally, the calibrated video camera can be used to measure the relative camera motion between successive frames. This chapter describes estimation of camera motion from video and the process of combining all of the derived sensor measurements such that the geoposition and orientation is precisely estimated at the time of each video frame acquisition.

The implemented estimation process is a recursive one that uses all of the measurements up to and including the current set of measurements to produce an estimate of the position and rotation of the camera relative to the WGS84 coordinate frame. Further, the sequential estimation process allows for both asynchronous measurements and unavailability of measurement. These properties are especially important to this application, as some measurements are not always available (e.g., GPS dropouts, corrupt video frames, etc.) and when measurements are available, they arrive at different frequencies. For example, the GPS receiver reports measurements at 1 Hz, while video frames are acquired at 14.34 Hz. The joint, sequential estimation process incorporates the uncertainties associated with each of these measurements to calculate the most probable geoposition and orientation with quantified uncertainty.

3.1 Sequential estimation of geoposition and orientation

Kalman filters are reliably used for estimating the translation and rotation of a calibrated camera from video acquired by the camera [90, 10, 53]. It is usual that this type of filter is applied to the problem of estimating the translation and rotation of the camera with respect to a relative coordinate system that is typically set to the coordinate frame of the camera at the time of the first video frame. This dissertation deals with the grander problem of estimating the camera position and orientation relative to an Earth-centered, Earth-fixed Cartesian coordinate system, namely the WGS84 geocentric coordinate frame. This section describes the Kalman filter used to sequentially estimate the camera geoposition and orientation from video-derived measurements as well as from positional measurements derived from the GPS receiver and rotational measurements from the 3-axis orientation sensor. Advantages of such a Kalman filter are that it incorporates multiple, independent measurements and that it explicitly estimates the camera position and orientation in the WGS84 geocentric coordinate frame.

The mapping of measurements of camera position and rotation, and their uncertainty, from a GPS receiver and 3-axis orientation sensor, respectively, is detailed in the previous chapter. Additional measurements of camera rotational and translational velocities may be derived by estimating the rotation and translation of the camera from the previous video frame to the current one. This is exactly the geometric relationship that is embodied by the essential matrix [47]. Estimation of the essential matrix between successive frames is described in section 3.1.1. Details of the Kalman filter follow in section 3.1.2.

3.1.1 Motion estimation from video

For each video frame, features are detected using the method described in [88]. This approach computes a 2×2 spatial gradient matrix for a specified window size about each pixel. The two eigenvalues of a spatial gradient matrix indicate the texturedness of the window: two small eigenvalues indicate a window of little texturedness (i.e., nearly constant intensity); one large eigenvalue, unidirectional texturedness; and two large eigenvalues, bidirectional texturedness (e.g., a corner). To mitigate the well-known aperture problem, only windows of bidirectional texturedness are selected for tracking. Windows containing this type of texturedness are determined by calcu-

lating the minor eigenvalue of each spatial gradient matrix and comparing the result to a predefined threshold. The feature detector selects windows with associated minor eigenvalues greater than the threshold value. Nonmaxima suppression is then applied to the remaining minor eigenvalues to limit the number of detected features.

A pyramidal implementation of the Lucas-Kanade feature tracker [52] then determines, for each feature, the translation from the previous frame to the current one. Central to this technique is a Newton-Raphson method of minimizing the differences between the window about the feature in the previous frame and the translated window in the current frame. The translation is iteratively estimated from the intensity difference between the two windows and the spatial gradients of the window in the current image. Further, multiresolution coarse-to-fine tracking allows for a larger window displacement from image to image while maintaining a smaller sized window, which is more reliably tracked. Figure 1.6 on page 18 shows detected and tracked features between successive video frames.

As described in the previous chapter, estimation of camera rotation and translation from images requires that image coordinates are mapped to non-distorted normalized coordinates using the inverse mapping procedure detailed on page 36. The resulting set of point correspondences in normalized coordinates may contain incorrect correspondences (e.g., due to erroneous feature tracking) that are inconsistent with the epipolar constraint of the essential matrix. Prior to estimation of the frame-to-frame camera rotation and translation, these incorrect correspondences are removed using a robust estimation method [80].

The Random Sample Consensus (RANSAC) algorithm [23] is a widely used robust estimator. RANSAC determines the largest consensus set within a set of correspondences that are consistent with the essential matrix using the following procedure.

1. Select a random sample of 7 correspondences and calculate the essential matrix. This will result in 1 or 3 solutions [53].

2. For each of the 1 or 3 solutions, determine the consensus set of correspondences that are within some tolerance of the so-called Sampson error of the points.

3. For each of the 1 or 3 consensus sets, if the size of the consensus set is greater than the stored largest consensus set (if any) then replace the stored largest consensus set with this consensus set.

4. If the number of trials to find the largest consensus set is reached, then terminate.

5. If the size of the largest consensus set is less than some threshold, then repeat; otherwise, terminate.

The resulting largest consensus set is the set of inliers that are consistent with the essential matrix. The remaining correspondences comprise the set of outliers. The Sampson error [81] is the distance between a measured point and its corresponding corrected point, where the corrected point is the first-order approximation of the closest point on the variety $\mathcal{V}_{\mathtt{E}}$ to the measured point (see [32] for details). In the case of the essential matrix $\mathtt{E}$, the squared Sampson error $\|\boldsymbol{\delta}_{(\hat{\hat{\mathbf{x}}}^\top, \hat{\hat{\mathbf{x}}}'^\top)}\|^2$ of the corresponding points $\hat{\hat{\mathbf{x}}} \leftrightarrow \hat{\hat{\mathbf{x}}}'$ is given by

$$\|\boldsymbol{\delta}_{(\hat{\hat{\mathbf{x}}}^\top, \hat{\hat{\mathbf{x}}}'^\top)}\|^2 = \frac{(\hat{\mathbf{x}}'^\top \mathtt{E} \hat{\mathbf{x}})^2}{(\hat{\mathbf{x}}'^\top \mathbf{e}_1)^2 + (\hat{\mathbf{x}}'^\top \mathbf{e}_2)^2 + (\mathbf{e}^{1\top} \hat{\mathbf{x}})^2 + (\mathbf{e}^{2\top} \hat{\mathbf{x}})^2}$$

where $\mathbf{e}_j$ is the jth column of $\mathtt{E}$, $\mathbf{e}^{i\top}$ is the ith row of $\mathtt{E}$, $\hat{\mathbf{x}} = (\hat{\hat{\mathbf{x}}}^\top, 1)^\top$, and $\hat{\mathbf{x}}' = (\hat{\hat{\mathbf{x}}}'^\top, 1)^\top$.

Finally, the essential matrix is estimated. The essential matrix $\mathtt{E} = [\mathbf{t}]_\times \exp(\boldsymbol{\omega})$ embodies the camera translation $\mathbf{t}$ and rotation $\boldsymbol{\omega}$, which have three degrees of freedom each. However, from a set of point correspondences, the essential matrix can only be determined to scale, i.e., the estimated essential matrix is a homogeneous entity. As such, it only has five degrees of freedom, which is insufficient to completely characterize $\mathbf{t}$ and $\boldsymbol{\omega}$. This constraint imposes that $\mathbf{t}$ can only be determined to scale, which indicates the direction of translation, but not the magnitude of the translation [32]. As with other homogeneous representations, it is convenient to constrain $\mathbf{t}$ such that $\|\mathbf{t}\| = 1$.

A specialized two-view bundle adjustment [93] process estimates the maximum likelihood estimate of the rotation and translation of the camera from the previous frame to the current one. It is specialized in the sense that the parameters of the camera associated with the previous frame are fixed to zero rotation and zero translation. The rotation $\boldsymbol{\omega}$ and translation $\mathbf{t}$ of the camera from the previous frame to the current one is computed using the Levenberg-Marquardt algorithm (see page 36). Throughout adjustment, $\mathbf{t}$ is constrained such that $\|\mathbf{t}\| = 1$ using the parameterization of the n-sphere [33, 71].

Applying Levenberg-Marquardt to this estimation problem, the measurement vector $\mathbf{X}$ is the set of n inlier point correspondences in normalized coordinates $\hat{\hat{\mathbf{x}}}_i \leftrightarrow \hat{\hat{\mathbf{x}}}'_i \; \forall \; i$,

$$\mathbf{X} = (\tilde{\mathbf{x}}_1^\top, \ldots, \tilde{\mathbf{x}}_n^\top, \tilde{\mathbf{x}}_1'^\top, \ldots, \tilde{\mathbf{x}}_n'^\top)^\top$$

with associated covariance matrix $\Sigma_{\mathbf{X}} = \text{diag}(\Sigma_{\tilde{\mathbf{x}}_1}, \ldots, \Sigma_{\tilde{\mathbf{x}}_n}, \Sigma_{\tilde{\mathbf{x}}'_1}, \ldots, \Sigma_{\tilde{\mathbf{x}}'_n})$. For clarity, the hat notation is removed from the points in normalized coordinates in the measurement vector $\mathbf{X}$ so that they are not confused with the estimate of the measurement vector $\widehat{\mathbf{X}}$. The parameter vector $\widehat{\mathbf{P}}$ is given by

$$\widehat{\mathbf{P}} = (\widehat{\boldsymbol{\omega}}^\top, \hat{\mathbf{t}}^\top, \widehat{\tilde{\mathbf{X}}}_1^\top, \ldots, \widehat{\tilde{\mathbf{X}}}_n^\top)^\top$$

where $\widehat{\tilde{\mathbf{X}}}_i \; \forall \; i$ is the set of pre-image 3D scene points. The algorithm iteratively finds the parameter vector $\widehat{\mathbf{P}}$ that minimizes the reprojection error $\boldsymbol{\epsilon}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = \mathbf{X} - \widehat{\mathbf{X}}$.

An estimate of the parameter vector $\widehat{\mathbf{P}}$ is mapped to an estimate of the measurement vector $\widehat{\mathbf{X}}$ using the equations

$$\hat{\mathbf{x}}_i = \hat{\mathtt{P}} \widehat{\mathbf{X}}_i \qquad\qquad \hat{\mathbf{x}}'_i = \hat{\mathtt{P}}' \widehat{\mathbf{X}}_i$$

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{w}_i \end{pmatrix} = [\mathtt{I} \mid \mathbf{0}] \begin{pmatrix} \widehat{\tilde{\mathbf{X}}}_i \\ 1 \end{pmatrix} \qquad \begin{pmatrix} \hat{x}'_i \\ \hat{y}'_i \\ \hat{w}'_i \end{pmatrix} = [\exp(\widehat{\boldsymbol{\omega}}) \mid \hat{\mathbf{t}}] \begin{pmatrix} \widehat{\tilde{\mathbf{X}}}_i \\ 1 \end{pmatrix}$$

Conversion to inhomogeneous points $\hat{\tilde{\mathbf{x}}}_i = (\hat{x}_i/\hat{w}_i, \hat{y}_i/\hat{w}_i)^\top$ and $\hat{\tilde{\mathbf{x}}}'_i = (\hat{x}'_i/\hat{w}'_i, \hat{y}'_i/\hat{w}'_i)^\top$ completes the mapping. As the Jacobian matrix $\mathtt{J} = \partial \widehat{\mathbf{X}} / \partial \widehat{\mathbf{P}}$ and subsequent matrices that operate on $\mathtt{J}$ contain a large number of zero elements, a more efficient sparse implementation of the Levenberg-Marquardt algorithm [34] is used.

Initial estimates of the rotation $\widehat{\boldsymbol{\omega}}$, translation $\hat{\mathbf{t}}$, and set of 3D points $\widehat{\tilde{\mathbf{X}}}_i \; \forall \; i$ contained in the parameter vector $\widehat{\mathbf{P}}$ are calculate as follows. From the set of inlier feature point correspondences, the Direct Linear Transformation (DLT) algorithm estimates the essential matrix $\mathtt{E}$, which is then decomposed to a rotation $\widehat{\boldsymbol{\omega}}$ and unit translation vector $\hat{\mathbf{t}}$ as described in [53]. Next, initial estimates of the 3D points are determined in two steps. First, corrected correspondences that minimize the geometric error subject to the epipolar constraint are calculated for all inlier feature correspondences. This is accomplished using the non-iterative, optimal method of [35]. Given the set of corrected correspondences, initial estimates of $\widehat{\tilde{\mathbf{X}}}_i \; \forall \; i$ are estimated by triangulation using the DLT algorithm as described in [32].

3.1.2 Kalman filter

The system uses an extended Kalman filter (EKF; an extension of the Kalman-Bucy filter [41, 8] to nonlinear systems) to estimate the geoposition and orientation of

the camera at each video frame n. In Kalman filter terminology, an EKF estimates the state $\mathbf{x}$ of a process that is governed by the nonlinear stochastic difference equation

$$\mathbf{x}_n = f(\mathbf{x}_{n-1}, \mathbf{u}_{n-1}, \mathbf{w}_{n-1})$$

where f is the nonlinear function that maps $\mathbf{x}$ from time step $n-1$ to time step n, $\mathbf{u}_{n-1}$ is the control input, and $\mathbf{w}_{n-1}$ is the process noise, which is unknown. The filter calculates state estimates $\hat{\mathbf{x}}$ from measurements. A measurement $\mathbf{z}_n$ at time step n is given by

$$\mathbf{z}_n = h(\mathbf{x}_n, \mathbf{v}_n)$$

where h is the nonlinear function that maps $\mathbf{x}_n$ to $\mathbf{z}_n$, and $\mathbf{v}_n$ is the measurement noise, which is also unknown.

Although the noises $\mathbf{w}$ and $\mathbf{v}$ are unknown at time step n, the state and measurement vectors at time step n can be approximated as

$$\begin{aligned} \hat{\mathbf{x}}_n^- &= f(\hat{\mathbf{x}}_{n-1}, \mathbf{u}_{n-1}, 0) \\ \hat{\mathbf{z}}_n^- &= h(\hat{\mathbf{x}}_n^-, 0) \end{aligned} \tag{3.1}$$

These approximations, or predictions, $\hat{\mathbf{x}}_n^-$ and $\hat{\mathbf{z}}_n^-$ are called the a priori estimates of the state and measurement vectors, respectively.

The a priori estimate of the state vector $\hat{\mathbf{x}}_n^-$ at time step n is given by (3.1). The covariance associated with the a priori state estimate, called the a priori state error covariance estimate $\mathtt{P}_n^-$, is given by

$$\mathtt{P}_n^- = \mathtt{A}_n \mathtt{P}_{n-1} \mathtt{A}_n^\top + \mathtt{W}_n \mathtt{Q}_{n-1} \mathtt{W}_n^\top \tag{3.2}$$

where the Jacobian matrices $\mathtt{A}_n$ and $\mathtt{W}_n$ are given by

$$\mathtt{A}_n = \frac{\partial \hat{\mathbf{x}}_n^-}{\partial \hat{\mathbf{x}}_{n-1}} \quad \text{and} \quad \mathtt{W}_n = \frac{\partial \hat{\mathbf{x}}_n^-}{\partial \mathbf{w}_{n-1}}$$

and $\mathtt{Q}_{n-1}$ is the process error covariance.

The filter then corrects the a priori estimates $\hat{\mathbf{x}}_n^-$ and $\mathtt{P}_n^-$ from a measurement $\hat{\mathbf{z}}_n$ at time step n. The corrected estimate and its associated covariance matrix are called the a posteriori state estimate $\hat{\mathbf{x}}_n$ and a posteriori error covariance $\mathtt{P}_n$. They are given by

$$\begin{aligned} \hat{\mathbf{x}}_n &= \hat{\mathbf{x}}_n^- + \mathtt{K}_n(\mathbf{z}_n - h(\hat{\mathbf{x}}_n^-, 0)) \\ \mathtt{P}_n &= (\mathtt{I} - \mathtt{K}_n \mathtt{H}_n)\mathtt{P}_n^- \end{aligned}$$

where $\mathtt{K}$ is the Kalman gain given by

$$\mathtt{K}_n = \mathtt{P}_n^- \mathtt{H}_n^\top (\mathtt{H}_n \mathtt{P}_n^- \mathtt{H}_n^\top + \mathtt{V}_n \mathtt{R}_n \mathtt{V}_n^\top)^{-1}$$

where $\mathtt{R}$ is measurement error covariance and the Jacobian matrices $\mathtt{H}_n$ and $\mathtt{V}_n$ are given by

$$\mathtt{H}_n = \frac{\partial \hat{\mathbf{z}}_n}{\partial \hat{\mathbf{x}}_n^-} \quad \text{and} \quad \mathtt{V}_n = \frac{\partial \hat{\mathbf{z}}_n}{\partial \mathbf{v}_n}$$

The Kalman filter is a recursive means of estimating the state of a process where the mean of the squared error is minimized.

We now describe application of an EKF to the estimation of the geoposition and orientation of a camera in WGS84 geocentric coordinates from video, GPS receiver measurements, and 3-axis orientation measurements. It is assumed that the rotational velocity $\dot{\boldsymbol{\omega}}$ and positional velocity $\dot{\mathbf{C}}$ of the camera are constant between successive video frames, where each frame is a time step in the filter. Due to the high frame rate of typical video cameras, this is a reasonable assumption. Under the constant velocity model, the state vector $\mathbf{x} = (\boldsymbol{\omega}^\top, \dot{\boldsymbol{\omega}}^\top, \mathbf{C}^\top, \dot{\mathbf{C}}^\top)^\top$, where $\mathbf{C}$ is the coordinates of the camera center in the WGS84 geocentric coordinate frame and $\boldsymbol{\omega}$ is the rotation from the WGS84 geocentric coordinate frame to the camera coordinate frame. For clarity the tilde has been removed from the camera center.

The initial state estimate $\hat{\mathbf{x}}_0$ is established from GPS receiver and 3-axis orientation sensor derived measurements such that $\widehat{\mathbf{C}}$ is set to the coordinates of the camera center in the WGS84 geocentric coordinates and $\widehat{\boldsymbol{\omega}}$ is set to the rotation that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame. Additionally, $\widehat{\dot{\boldsymbol{\omega}}}$ is set to $\mathbf{0}$ and $\widehat{\dot{\mathbf{C}}}$ is set to a random vector with mean $\mathbf{0}$ and covariance $\Sigma = \mathrm{diag}(\sigma^2, \sigma^2, \sigma^2)$, where σ is a conservative estimate of the standard deviation of the positional velocity of the camera.

For the time update component of the filter, the entries in the a priori state estimate $\hat{\mathbf{x}}_n^-$ are given by

$$\begin{aligned}
\widehat{\boldsymbol{\omega}}_n^- &= \log(\exp(\widehat{\dot{\boldsymbol{\omega}}}_{n-1}) \exp(\widehat{\boldsymbol{\omega}}_{n-1})) \\
\widehat{\dot{\boldsymbol{\omega}}}_n^- &= \widehat{\dot{\boldsymbol{\omega}}}_{n-1} \\
\widehat{\mathbf{C}}_n^- &= \widehat{\mathbf{C}}_{n-1} + \widehat{\dot{\mathbf{C}}}_{n-1} \\
\widehat{\dot{\mathbf{C}}}_n^- &= \widehat{\dot{\mathbf{C}}}_{n-1}
\end{aligned}$$

and the Jacobian matrix $\mathtt{A}_n$ is given by

$$\mathtt{A}_n = \frac{\partial \hat{\mathbf{x}}_n^-}{\partial \hat{\mathbf{x}}_{n-1}} = \begin{bmatrix} \frac{\partial \widehat{\boldsymbol{\omega}}_n^-}{\partial \widehat{\boldsymbol{\omega}}_{n-1}} & \frac{\partial \widehat{\boldsymbol{\omega}}_n^-}{\partial \hat{\widehat{\boldsymbol{\omega}}}_{n-1}} & \mathtt{0} & \mathtt{0} \\ \mathtt{0} & \mathtt{I} & \mathtt{0} & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \frac{\partial \widehat{\mathbf{C}}_n^-}{\partial \widehat{\mathbf{C}}_{n-1}} & \frac{\partial \widehat{\mathbf{C}}_n^-}{\partial \hat{\widehat{\mathbf{C}}}_{n-1}} \\ \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{I} \end{bmatrix}$$

where

$$\frac{\partial \widehat{\boldsymbol{\omega}}_n^-}{\partial \widehat{\boldsymbol{\omega}}_{n-1}} = \frac{\partial \widehat{\boldsymbol{\omega}}_n^-}{\partial \hat{\mathbf{r}}_n^-} \frac{\partial \hat{\mathbf{r}}_n^-}{\partial \hat{\mathbf{r}}_{n-1}} \frac{\partial \hat{\mathbf{r}}_{n-1}}{\partial \widehat{\boldsymbol{\omega}}_{n-1}} \quad \text{and} \quad \frac{\partial \widehat{\boldsymbol{\omega}}_n^-}{\partial \hat{\widehat{\boldsymbol{\omega}}}_{n-1}} = \frac{\partial \widehat{\boldsymbol{\omega}}_n^-}{\partial \hat{\mathbf{r}}_n^-} \frac{\partial \hat{\mathbf{r}}_n^-}{\partial \hat{\hat{\mathbf{r}}}_{n-1}} \frac{\partial \hat{\hat{\mathbf{r}}}_{n-1}}{\partial \hat{\widehat{\boldsymbol{\omega}}}_{n-1}}$$

where $\partial \hat{\hat{\mathbf{r}}}_{n-1} / \partial \hat{\widehat{\boldsymbol{\omega}}}_{n-1}$, $\partial \hat{\mathbf{r}}_{n-1} / \partial \widehat{\boldsymbol{\omega}}_{n-1}$, and $\partial \widehat{\boldsymbol{\omega}}_n^- / \partial \hat{\mathbf{r}}_n^-$ are given by (2.9) and (2.10), and $\partial \hat{\mathbf{r}}_n^- / \partial \hat{\mathbf{r}}_{n-1}$, $\partial \hat{\mathbf{r}}_n^- / \partial \hat{\hat{\mathbf{r}}}_{n-1}$, $\partial \widehat{\mathbf{C}}_n^- / \partial \widehat{\mathbf{C}}_{n-1}$, and $\partial \widehat{\mathbf{C}}_n^- / \partial \hat{\widehat{\mathbf{C}}}_{n-1}$ are given by equations found in section A.1 on page 78.

Measurement updates

The a priori state estimate $\hat{\mathbf{x}}_n^-$ is corrected by three potential measurements: the camera position derived from the GPS receiver measurements, camera rotation derived from the 3-axis orientation measurements, or camera rotational velocity derived from video. Additionally, if a GPS receiver and 3-axis orientation measurement is received at the same time step, this will result in correlated camera position and rotation measurements, which has a separate set of update equations. Note that camera positional velocity measurements are not derived from video. This is due to the fact that the camera either may not translate between successive frames or translate by such a small magnitude, that the translation estimate is erroneous due to noise. However, it has been shown through experimentation that camera rotation is correctly estimated, despite an incorrect translation estimate [90, 53].

If multiple uncorrelated measurements (e.g., camera position via the GPS receiver and camera rotational velocity from video) are received at the same time step n, each measurement $\mathbf{z}_n^{(j)}$ for $j = 1, \ldots, m$ is used to estimate the state as follows. At the beginning of time step n, the filter performs a time update to calculate the intermediate a priori state estimate $\hat{\mathbf{x}}_n^{(1)-}$. Next, the first measurement $\mathbf{z}_n^{(1)}$ is used to correct the intermediate a priori estimate $\hat{\mathbf{x}}_n^{(1)-}$, yielding the intermediate a posteriori estimate $\hat{\mathbf{x}}_n^{(1)}$. This is followed by a second time update, but with a time step size of zero, producing the second intermediate a priori state estimate $\hat{\mathbf{x}}_n^{(2)-}$. However, a step size of zero is

equivalent to not performing a time update, so $\hat{\mathbf{x}}_n^{(2)-} = \hat{\mathbf{x}}_n^{(1)}$. The second intermediate a priori estimate $\hat{\mathbf{x}}_n^{(2)-} = \hat{\mathbf{x}}_n^{(1)}$ is corrected by the second measurement $\mathbf{z}_n^{(2)}$, yielding the second intermediate a posteriori estimate and third intermediate a priori estimate $\hat{\mathbf{x}}_n^{(2)} = \hat{\mathbf{x}}_n^{(3)-}$. This is continued for all m uncorrelated measurements at time step n. The intermediate a priori estimate $\hat{\mathbf{x}}_n^{(m)}$ due to the last measurement $\mathbf{z}_n^{(m)}$ is the final a priori estimate $\hat{\mathbf{x}}_n = \mathbf{z}_n^{(m)}$ at time step n.

When the GPS receiver reports a new measurement, it is immediately converted to WGS84 geocentric coordinates $\mathbf{C}$ with associated covariance matrix $\Sigma_{\mathbf{C}}$ as detailed in section 2.2 on page 39. If the 3-axis orientation sensor also reports a measurement in the same time step, then the origin of the camera coordinate frame is set to $\mathbf{C}$. As such, the calculated rotation $\boldsymbol{\omega}$ that maps coordinates in the WGS84 geocentric coordinate frame to coordinates in the camera coordinate frame is correlated to $\mathbf{C}$ and the covariances of $(\theta, \psi, \phi)^\top$ and $\mathbf{C}$ are jointly propagated to the covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{C}^\top)}$ (see section 2.3 on page 44). In this case, the a priori state estimate is corrected by the measurement $\mathbf{z}_n = (\boldsymbol{\omega}^\top, \mathbf{C}^\top)^\top$ with associated covariance matrix $\mathtt{R}_n = \Sigma_{(\boldsymbol{\omega}^\top, \mathbf{C}^\top)}$ and the Jacobian matrix $\mathtt{H}_n$ is given by

$$\mathtt{H}_n = \frac{\partial \hat{\mathbf{z}}_n}{\partial \hat{\mathbf{x}}_n^-} = \frac{\partial (\hat{\boldsymbol{\omega}}_n^\top, \widehat{\mathbf{C}}_n^\top)}{\partial \hat{\mathbf{x}}_n^-} = \begin{bmatrix} \mathtt{I} & \mathtt{0} & \mathtt{0} & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \mathtt{I} & \mathtt{0} \end{bmatrix}$$

Otherwise, if a 3-axis orientation sensor measurement is not reported, then only the GPS derived measurement of the camera center is used to update the a priori state estimate using the measurement $\mathbf{z}_n = \mathbf{C}$ with covariance $\mathtt{R}_n = \Sigma_{\mathbf{C}}$ and the Jacobian matrix $\mathtt{H}_n$ is given by

$$\mathtt{H}_n = \frac{\partial \hat{\mathbf{z}}_n}{\partial \hat{\mathbf{x}}_n^-} = \frac{\partial \widehat{\mathbf{C}}_n}{\partial \hat{\mathbf{x}}_n^-} = \begin{bmatrix} \mathtt{0} & \mathtt{0} & \mathtt{I} & \mathtt{0} \end{bmatrix}$$

In the case of a reported 3-axis orientation sensor measurement, but not a GPS receiver measurement, the origin of the camera coordinate frame is set to the a priori estimate of the camera center $\widehat{\mathbf{C}}_n^-$ where the associated covariance $\Sigma_{\widehat{\mathbf{C}}_n^-}$ is the 3×3 block on the diagonal of the a priori state error covariance estimate $\mathtt{P}_n^-$ corresponding to $\widehat{\mathbf{C}}_n^-$. Similar to above, the rotation $\boldsymbol{\omega}$ calculated from the pitch, roll, and yaw $(\theta, \psi, \phi)^\top$ measurements is correlated to $\widehat{\mathbf{C}}_n^-$, and the covariances of $(\theta, \psi, \phi)^\top$ and $\widehat{\mathbf{C}}_n^-$ are jointly propagated to the covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \widehat{\mathbf{C}}_n^{-\top})}$. However, $\widehat{\mathbf{C}}_n^-$ is not a measurement and, as such, is not used to correct the a priori state estimate $\hat{\mathbf{x}}_n^-$ (i.e., it is not used to correct itself). Only the derived measurement of the rotation is used to update the

a priori state estimate. Specifically, the measurement $\mathbf{z}_n = \boldsymbol{\omega}$ with covariance $\mathtt{R}_n = \Sigma_{\boldsymbol{\omega}}$ and the Jacobian matrix $\mathtt{H}_n$ is given by

$$\mathtt{H}_n = \frac{\partial \hat{\mathbf{z}}_n}{\partial \hat{\mathbf{x}}_n^-} = \frac{\partial \widehat{\boldsymbol{\omega}}_n}{\partial \hat{\mathbf{x}}_n^-} = \begin{bmatrix} \mathtt{I} & \mathtt{0} & \mathtt{0} & \mathtt{0} \end{bmatrix}$$

The last potential measurement is that of the camera rotational velocity $\dot{\boldsymbol{\omega}}$. A measure of the rotational velocity is the estimate of the rotation of the camera from the previous frame $n-1$ to the current one n as described in section 3.1.1. For the Kalman filter update equations, $\mathbf{z}_n = \dot{\boldsymbol{\omega}}$, $\mathtt{R}_n = \Sigma_{\dot{\boldsymbol{\omega}}}$, and the Jacobian matrix $\mathtt{H}_n$ is given by

$$\mathtt{H}_n = \frac{\partial \hat{\mathbf{z}}_n}{\partial \hat{\mathbf{x}}_n^-} = \frac{\partial \widehat{\dot{\boldsymbol{\omega}}}_n}{\partial \hat{\mathbf{x}}_n^-} = \begin{bmatrix} \mathtt{0} & \mathtt{I} & \mathtt{0} & \mathtt{0} \end{bmatrix}$$

4

Multiple Camera Estimation

This chapter extends the work of the previous chapter from estimation of the geoposition and orientation of a single camera to that of multiple cameras. Specifically, in the case of multiple cameras imaging the same region of a scene, these independent observations of the features in the scene are used to further refine the geoposition and orientation of the cameras, provided that feature correspondences are established between the images acquired from different cameras.

Central to this chapter is determination of the search region in one image that, at some specified probability, contains an image feature that corresponds to a feature in another image. A general approach and analytical method for determining a search region for use in guided matching under projective mappings is developed [72]. This method is used to guide a feature in one image to its corresponding feature in another image, dictated by the relative imaging geometry of the cameras that acquired the images.

The remainder of this chapter includes feature detection and comparison for matching, as well as the process of jointly estimating the maximum likelihood of the geoposition and orientation of all cameras imaging the same region of a scene for which feature correspondences have been established.

4.1 Feature detection and matching

Recent work has shown that distinct image features that are invariant to viewpoint and illumination changes can be reliably detected [51, 63]. These types of changes

are locally modeled as an affinity or similarity (affinity minus skew). Examples of such feature detectors include ones based on affine normalization and Hessian points [61, 82], the Maximally Stable Extremal Region (MSER) detector [58], detectors based on edges and intensity extrema [96, 95], one that detects salient regions [40], and the Scale Invariant Feature Transform (SIFT) detector [49, 50, 51]. An affinity is sufficient to locally model geometric distortions arising from viewpoint changes provided that the local neighborhood about the scene feature can be approximated by a plane. Although a similarity does not model skew, it has been shown to perform well in similar applications, such as robotics [83, 84]. It is also assumed that photometric deformations can be modeled by a linear transformation of the local intensities. In this dissertation, image features are detected using the SIFT detector. Examples of regions detected by the SIFT detector are shown in figure 1.7 on page 23.

For each detected region, a local description of the intensity pattern within the region is calculated. The feature matching process, described in section 4.1.2 on page 71, uses these local descriptors to determine the similarity between different features. A recent comparison of local descriptors [62] indicates that SIFT descriptors, each typically a 128-dimensional vector representing a local image region sampled relative to its scale-space coordinate frame, are superior to other descriptors. Further, the vector is organized such that the Euclidean distance between any two SIFT descriptor vectors is a measure of the similarity between the SIFT features described by the vectors, i.e., smaller distances are more similar. The work presented here uses the SIFT reference implementation [48] for both feature detection and calculation of the local descriptor.

The remainder of this section addresses the problem of matching the detected features across images acquired from different cameras that are imaging the same region of a scene. Focus is given to determination of the region in an image to search for a corresponding feature—the guided matching problem. Other components of the matching process are feature comparison to establish an initial set of correspondences followed by robust outlier rejection.

4.1.1 Covariance propagation for guided matching

In this work we address the problem of determining a search region used for establishing feature correspondences over multiple views given an estimate of the projective mapping that relates these views. Corresponding features are defined as the set

of features that are the images of the same pre-image feature. Consider two different cameras imaging a scene. A 3D point in the scene is imaged as a 2D point in the image plane of each of the cameras. The image point in one of the cameras corresponds to the image point in the other camera and both image points correspond to the pre-image 3D scene point. Several projective models (e.g., the fundamental matrix) have been developed in computer vision that allow features to be mapped between views without explicit knowledge of the 3D scene structure. However, in the presence of noise or uncertainty, the mapped feature may not be coincident with the true corresponding feature and a search must be performed to locate the true correspondence. In the absence of uncertainty information, the true correspondence may be located anywhere in the image, assuming the pre-image feature was imaged by the camera.

Guided matching methods are often used to reduce the size of the search region from the entire image to a region expected to contain the corresponding feature. One simple guided matching method is to specify a search region bounded at a fixed distance from the mapped feature. Although easy to implement, this simple method generally yields either an undersized region, which may not include the true correspondence, or an oversized region, which may include features that are similar to the true correspondence, increasing the potential of a false match. Our approach uses covariance propagation to define the search region for projective mappings. The search region is bounded by a specified probability that the region contains the feature. This approach can be used, for example, to propagate the spatial covariance of a homogeneous point through a planar projective (homography) or epipolar (fundamental matrix) transformation, where the transformation may additionally have an associated covariance. We present an expression that allows for the determination of the covariance of the mapping of the point as a function of the above covariances.

Several approaches to propagating uncertainty in structure from motion have been proposed appealing to different statistical techniques. One approach is to use Monte Carlo methods, which are highly general but computationally expensive. Alternatively, analytical frameworks have been developed by Kanatani [42], Förstner [25], and others (e.g., [85], [92]). However, when these mappings are well approximated locally by an affine transform, a first-order model has proved to be sufficient [32]. This linearized approximation of the error model is commonly used in computer vision and is the approach adopted in this paper. Central to this approach to uncertainty propagation is

the Jacobian matrix of the mapping. One method for estimating the Jacobian is to perform numerical differentiation using forward differencing, which in practice may yield a Jacobian matrix of incorrect rank due to numerical inaccuracies. Alternatively, one can derive specialized analytical expressions on a per-mapping basis, e.g., planar homography [14] and fundamental matrix [15], [99]. We derive a novel analytical expression for the Jacobian applicable to all projective mappings, obviating the need for specialized expressions. The resulting closed-form expression is general and easy to implement. The same expression can be generalized to n dimensions and can also be applied to other projective mappings such as composition of homographies.

Nonlinear propagation of covariance

Let $\mathbf{x} \in \mathbb{R}^n$ be a random vector with mean $\boldsymbol{\mu}_\mathbf{x}$ and covariance matrix $\Sigma_\mathbf{x}$, and let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a nonlinear function. Up to first-order approximation, $\mathbf{y} = f(\mathbf{x}) \approx f(\boldsymbol{\mu}_\mathbf{x}) + \mathtt{J}(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})$, where $\mathtt{J} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix $\partial f / \partial \mathbf{x}$ evaluated at $\boldsymbol{\mu}_\mathbf{x}$. If f is approximately affine in the region about the mean of the distribution, then this approximation is reasonable and the random vector $\mathbf{y} \in \mathbb{R}^m$ has mean $\boldsymbol{\mu}_\mathbf{y} \approx f(\boldsymbol{\mu}_\mathbf{x})$ and covariance $\Sigma_\mathbf{y} \approx \mathtt{J}\Sigma_\mathbf{x}\mathtt{J}^\top$.

If $\mathbf{x}$ is composed of two random vectors $\mathbf{a}$ and $\mathbf{b}$ such that $\mathbf{x} = (\mathbf{a}^\top, \mathbf{b}^\top)^\top$, then

$$\Sigma_\mathbf{y} \approx \begin{bmatrix} \mathtt{J}_\mathbf{a} & \mathtt{J}_\mathbf{b} \end{bmatrix} \begin{bmatrix} \Sigma_\mathbf{a} & \Sigma_\mathbf{ab} \\ \Sigma_\mathbf{ba} & \Sigma_\mathbf{b} \end{bmatrix} \begin{bmatrix} \mathtt{J}_\mathbf{a}^\top \\ \mathtt{J}_\mathbf{b}^\top \end{bmatrix} \tag{4.1}$$

where $\mathtt{J}_\mathbf{a} = \partial \mathbf{y} / \partial \mathbf{a}$ and $\mathtt{J}_\mathbf{b} = \partial \mathbf{y} / \partial \mathbf{b}$.

Projective mappings Consider a homogeneous 2D point $\mathbf{x}$ represented by the vector $(x, y, w)^\top \in \mathbb{R}^3$. The vector $s(x, y, w)^\top$, where s is any nonzero scalar, represents the same 2D point as $(x, y, w)^\top$. It follows that $(x, y, w)^\top \sim s(x, y, w)^\top$, where $\sim$ denotes equality up to a nonzero scale factor.

Due to the use of homogeneous representations, several projective mappings are only determined up to scale [86]. Examples include point imaging $\mathbf{x} \sim \mathtt{P}\mathbf{X}$, where $\mathtt{P} \in \mathbb{R}^{3 \times 4}$ is the projective camera that maps the homogeneous 3D point $\mathbf{X}$ to a homogeneous 2D point $\mathbf{x}$; projective transformation accumulation $\mathtt{H}_{a,c} \sim \mathtt{H}_{b,c}\mathtt{H}_{a,b}$, where $\mathtt{H}_{a,c}, \mathtt{H}_{b,c}, \mathtt{H}_{a,b} \in \mathbb{R}^{(n+1) \times (n+1)}$ are n-dimensional projective transformations and $\mathtt{H}_{a,c}$ represents the transformation from a to c; and point-to-line mapping $\boldsymbol{\ell}' \sim \mathtt{F}\mathbf{x}$, where $\mathtt{F} \in \mathbb{R}^{3 \times 3}$

is the fundamental matrix, which maps a homogeneous 2D point $\mathbf{x}$ in one image to a homogeneous 2D line $\boldsymbol{\ell}'$ in another image. All of these mappings can be generalized as $\mathtt{C} \sim \mathtt{AB}$, where $\mathtt{C} \in \mathbb{R}^{m\times n}$, $\mathtt{A} \in \mathbb{R}^{m\times p}$, and $\mathtt{B} \in \mathbb{R}^{p\times n}$. However, because $\mathtt{C}$ is only determined up to scale, the entries of $\mathtt{C}$ may vary without bound. This poses an issue for covariance propagation and uncertainty analysis. It is usual to impose the constraint that $\|\mathtt{C}\| = 1$, where $\|\cdot\|$ denotes the Frobenius norm. Under this constraint, the generalized mapping is $\mathtt{C} = (\mathtt{AB})/\|\mathtt{AB}\|$ and the variance of the entries of $\mathtt{C}$ are constrained accordingly.

Jacobian matrices For the expression

$$\mathtt{C} = \frac{\mathtt{AB}}{\|\mathtt{AB}\|}$$

analytical derivations of the Jacobian matrices $\partial\mathbf{c}/\partial\mathbf{a}$ and $\partial\mathbf{c}/\partial\mathbf{b}$ are as follows.

Assume $\mathtt{A} \in \mathbb{R}^{m\times p}$, $\mathtt{B} \in \mathbb{R}^{p\times n}$, and $\mathtt{C} \in \mathbb{R}^{m\times n}$. For the equation

$$\mathtt{C} = \frac{\mathtt{AB}}{\|\mathtt{AB}\|} = \frac{\mathtt{M}}{\|\mathtt{M}\|}$$

where $\|\cdot\|$ denotes the Frobenius norm, we seek the partial derivatives of the entries of $\mathtt{C}$ with respect to the entries of both $\mathtt{A}$ and $\mathtt{B}$ [31]. For clarity, the matrix product $\mathtt{AB}$ is denoted by $\mathtt{M}$, which allows for $\mathrm{vec}((\mathtt{AB})^\top)$ to be represented by $\mathbf{m}$. The partial derivative of $\mathbf{c}$ with respect to $\mathbf{a}$ is computed as

$$\frac{\partial\mathbf{c}}{\partial\mathbf{a}} = \frac{1}{\|\mathbf{m}\|^2}\left[\|\mathbf{m}\|\frac{\partial\mathbf{m}}{\partial\mathbf{a}} - \mathbf{m}\frac{\partial\|\mathbf{m}\|}{\partial\mathbf{a}}\right]$$

or equivalently

$$\frac{\partial\mathbf{c}}{\partial\mathbf{a}} = \frac{1}{\|\mathbf{m}\|}\left[\frac{\partial\mathbf{m}}{\partial\mathbf{a}} - \mathbf{c}\frac{\partial\|\mathbf{m}\|}{\partial\mathbf{a}}\right] \tag{4.2}$$

Similarly,

$$\frac{\partial\mathbf{c}}{\partial\mathbf{b}} = \frac{1}{\|\mathbf{m}\|}\left[\frac{\partial\mathbf{m}}{\partial\mathbf{b}} - \mathbf{c}\frac{\partial\|\mathbf{m}\|}{\partial\mathbf{b}}\right] \tag{4.3}$$

The partial derivative of $\mathbf{m}$ with respect to both $\mathbf{a}$ and $\mathbf{b}$ is given by

$$\frac{\partial\mathbf{m}}{\partial\mathbf{a}} = \mathtt{I}_{m\times m} \otimes \mathtt{B}^\top \text{and } \frac{\partial\mathbf{m}}{\partial\mathbf{b}} = \mathtt{A} \otimes \mathtt{I}_{n\times n}$$

where $\otimes$ denotes the Kronecker product. The partial derivative of $\|\mathtt{M}\|$ with respect to both $\mathtt{A}$ and $\mathtt{B}$ is

$$\frac{\partial\|\mathtt{M}\|}{\partial\mathtt{A}} = \mathtt{C}\mathtt{B}^\top \text{and } \frac{\partial\|\mathtt{M}\|}{\partial\mathtt{B}} = \mathtt{A}^\top\mathtt{C}$$

It follows that

$$\frac{\partial \|\mathbf{m}\|}{\partial \mathbf{a}} = \text{vec}(\mathtt{B}\mathtt{C}^\top)^\top \text{and } \frac{\partial \|\mathbf{m}\|}{\partial \mathbf{b}} = \text{vec}(\mathtt{C}^\top \mathtt{A})^\top$$

Substituting into (4.2) and (4.3) yields

$$\mathtt{J}_\mathbf{a} = \frac{\partial \mathbf{c}}{\partial \mathbf{a}} = \frac{1}{\|\mathtt{A}\mathtt{B}\|}\left[\left(\mathtt{I}_{m\times m} \otimes \mathtt{B}^\top\right) - \mathbf{c}\,\text{vec}(\mathtt{B}\mathtt{C}^\top)^\top\right] \tag{4.4}$$

$$\mathtt{J}_\mathbf{b} = \frac{\partial \mathbf{c}}{\partial \mathbf{b}} = \frac{1}{\|\mathtt{A}\mathtt{B}\|}\left[(\mathtt{A} \otimes \mathtt{I}_{n\times n}) - \mathbf{c}\,\text{vec}(\mathtt{C}^\top \mathtt{A})^\top\right] \tag{4.5}$$

Applying (4.1), these are the Jacobian matrices used to approximate the covariance $\Sigma_\mathbf{c}$.

Guided matching

Search region If a Gaussian random vector $\mathbf{x}$ has mean $\boldsymbol{\mu}_\mathbf{x}$ and covariance $\Sigma_\mathbf{x}$, then the squared Mahalanobis distance between $\mathbf{x}$ and $\boldsymbol{\mu}_\mathbf{x}$ satisfies a χ^2_r distribution where r is the degrees of freedom of $\mathbf{x}$. It follows that a percentage α of all instances of $\mathbf{x}$ will satisfy the condition

$$(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \Sigma_\mathbf{x}^+ (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) \leq k^2 \tag{4.6}$$

where k^2 is the inverse of the chi-square cumulative distribution function with r degrees of freedom and probability α, and $\Sigma_\mathbf{x}^+$ is the pseudo-inverse of the covariance matrix $\Sigma_\mathbf{x}$ with rank r.

2D points and lines A derivation of the uncertainty bounds for homogeneous 2D lines may be found in [15], [99], and [32]. In this section we derive uncertainty bounds for 2D points and state the bounds for 2D lines using the duality principle.

Let $\boldsymbol{\mu}_\mathbf{x}$ and $\Sigma_\mathbf{x}$ be the mean and covariance, respectively, of a homogeneous 2D point $\mathbf{x}$. The covariance matrix has rank 2, thereby constraining the 3-vector to 2 degrees of freedom. For a given k, we can determine if an instance of $\mathbf{x}$ is within the uncertainty bounds directly from (4.6). However, it is often the case that we want to determine if an *arbitrary* homogeneous point, not necessarily drawn from the distribution of $\mathbf{x}$, is within these bounds. This cannot be accomplished using (4.6) directly and instead must be determined geometrically as follows.

For a given k, the set of points with equal likelihood in the distribution of $\mathbf{x}$ is given by

$$(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \Sigma_\mathbf{x}^+ (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) = k^2 \tag{4.7}$$

For further analysis, we apply a change of coordinates such that

$$\Sigma'_{\mathbf{x}} = \mathtt{U}\Sigma_{\mathbf{x}}\mathtt{U}^\top = \begin{bmatrix} \tilde{\Sigma}'_{\mathbf{x}} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}$$

where $\tilde{\Sigma}'_{\mathbf{x}} \in \mathbb{R}^{2\times 2}$ is a nonsingular diagonal matrix, and $\boldsymbol{\mu}'_{\mathbf{x}} = \mathtt{U}\boldsymbol{\mu}_{\mathbf{x}} = (\tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}}, 1)^\top$ and $\mathbf{x}' = \mathtt{U}\mathbf{x} = (\tilde{\mathbf{x}}'^\top, 1)^\top$. The similarity $\mathtt{U} = s\mathtt{V}^\top$, where the orthogonal matrix $\mathtt{V}^\top$ is obtained from the eigen decomposition $\Sigma_{\mathbf{x}} = \mathtt{V}\mathtt{D}\mathtt{V}^\top$ and s is chosen such that the last entry in the 3-vector $s\mathtt{V}^\top\boldsymbol{\mu}_{\mathbf{x}}$ is equal to 1. The matrix $\mathtt{D} = \mathrm{diag}(\lambda_1, \lambda_2, 0)$ contains the eigenvalues of $\Sigma_{\mathbf{x}}$. Using this, we can show

$$\begin{aligned} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})^\top \Sigma^+_{\mathbf{x}} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}) &= k^2 \\ (\mathbf{x}' - \boldsymbol{\mu}'_{\mathbf{x}})^\top \Sigma'^+_{\mathbf{x}} (\mathbf{x}' - \boldsymbol{\mu}'_{\mathbf{x}}) &= k^2 \\ (\tilde{\mathbf{x}}' - \tilde{\boldsymbol{\mu}}'_{\mathbf{x}})^\top \tilde{\Sigma}'^{-1}_{\mathbf{x}} (\tilde{\mathbf{x}}' - \tilde{\boldsymbol{\mu}}'_{\mathbf{x}}) &= k^2 \end{aligned} \tag{4.8}$$

which can be written more fully as

$$\tilde{\mathbf{x}}'^\top \tilde{\Sigma}'^{-1}_{\mathbf{x}} \tilde{\mathbf{x}}' - \tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}} \tilde{\Sigma}'^{-1}_{\mathbf{x}} \tilde{\mathbf{x}}' - \tilde{\mathbf{x}}'^\top \tilde{\Sigma}'^{-1}_{\mathbf{x}} \tilde{\boldsymbol{\mu}}'_{\mathbf{x}} + \tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}} \tilde{\Sigma}'^{-1}_{\mathbf{x}} \tilde{\boldsymbol{\mu}}'_{\mathbf{x}} - k^2 = 0$$

or in matrix form

$$\begin{pmatrix} \tilde{\mathbf{x}}'^\top & 1 \end{pmatrix} \begin{bmatrix} \tilde{\Sigma}'^{-1}_{\mathbf{x}} & -\tilde{\Sigma}'^{-1}_{\mathbf{x}} \tilde{\boldsymbol{\mu}}'_{\mathbf{x}} \\ -\tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}} \tilde{\Sigma}'^{-1}_{\mathbf{x}} & \tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}} \tilde{\Sigma}'^{-1}_{\mathbf{x}} \tilde{\boldsymbol{\mu}}'_{\mathbf{x}} - k^2 \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{x}}' \\ 1 \end{pmatrix} = 0$$

This is equivalent to

$$\begin{aligned} \begin{pmatrix} \tilde{\mathbf{x}}'^\top & 1 \end{pmatrix} \begin{bmatrix} \tilde{\boldsymbol{\mu}}'_{\mathbf{x}} \tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}} - k^2 \tilde{\Sigma}'_{\mathbf{x}} & \tilde{\boldsymbol{\mu}}'_{\mathbf{x}} \\ \tilde{\boldsymbol{\mu}}'^\top_{\mathbf{x}} & 1 \end{bmatrix}^{-1} \begin{pmatrix} \tilde{\mathbf{x}}' \\ 1 \end{pmatrix} &= 0 \\ \mathbf{x}'^\top \left[\boldsymbol{\mu}'_{\mathbf{x}} \boldsymbol{\mu}'^\top_{\mathbf{x}} - k^2 \Sigma'_{\mathbf{x}} \right]^{-1} \mathbf{x}' &= 0 \end{aligned}$$

which is the equation of a conic. The conic $\mathtt{C}' = [\boldsymbol{\mu}'_{\mathbf{x}} \boldsymbol{\mu}'^\top_{\mathbf{x}} - k^2 \Sigma'_{\mathbf{x}}]^{-1}$ is formed by the points that satisfy (4.8). Transforming back to the original coordinate system, $\mathtt{C} = \mathtt{U}^\top \mathtt{C}' \mathtt{U}$, the set of equal-likelihood points that satisfy (4.7) form the homogeneous conic

$$\mathtt{C} = \left[\boldsymbol{\mu}_{\mathbf{x}} \boldsymbol{\mu}^\top_{\mathbf{x}} - k^2 \Sigma_{\mathbf{x}} \right]^{-1} \tag{4.9}$$

representing an ellipse containing $\boldsymbol{\mu}_{\mathbf{x}}$. An arbitrary point $\mathbf{x}_0$ is on the interior of the ellipse if $\mathbf{x}_0^\top \mathtt{C} \mathbf{x}_0$ has the same sign as $\boldsymbol{\mu}^\top_{\mathbf{x}} \mathtt{C} \boldsymbol{\mu}_{\mathbf{x}}$.

Using the duality between points and lines, and conics and dual conics, the same approach is employed for homogeneous 2D lines. The set of equal-likelihood lines in the distribution of a random homogeneous line $\boldsymbol{\ell}$ with mean $\boldsymbol{\mu}_{\boldsymbol{\ell}}$ and covariance $\Sigma_{\boldsymbol{\ell}}$ satisfies

$$(\boldsymbol{\ell} - \boldsymbol{\mu}_{\boldsymbol{\ell}})^\top \Sigma_{\boldsymbol{\ell}}^{+} (\boldsymbol{\ell} - \boldsymbol{\mu}_{\boldsymbol{\ell}}) = k^2$$

for a given k. The set of lines form the homogeneous dual conic $\mathtt{C}^* = \left[\boldsymbol{\mu}_{\boldsymbol{\ell}}\boldsymbol{\mu}_{\boldsymbol{\ell}}^\top - k^2\Sigma_{\boldsymbol{\ell}}\right]^{-1}$, which is the adjoint of the matrix $\mathtt{C}$. Therefore, for a non-singular symmetric matrix $\mathtt{C} \sim (\mathtt{C}^*)^{-1}$, the conic that forms the envelope of lines is given by

$$\mathtt{C} = \boldsymbol{\mu}_{\boldsymbol{\ell}}\boldsymbol{\mu}_{\boldsymbol{\ell}}^\top - k^2\Sigma_{\boldsymbol{\ell}} \tag{4.10}$$

This conic is a hyperbola with branches symmetric about $\boldsymbol{\mu}_{\boldsymbol{\ell}}$. An arbitrary point $\mathbf{x}_0$ lies inside the region between the two branches of the hyperbola if $\mathbf{x}_0^\top \mathtt{C}\mathbf{x}_0$ has the same sign as $\mathbf{x}_{\boldsymbol{\ell}}^\top \mathtt{C}\mathbf{x}_{\boldsymbol{\ell}}$, where $\mathbf{x}_{\boldsymbol{\ell}}$ is any point that lies on the line $\boldsymbol{\mu}_{\boldsymbol{\ell}}$. Two points $\mathbf{x}_1$ and $\mathbf{x}_2$ on the line $\boldsymbol{\mu}_{\boldsymbol{\ell}}$ may be determined by $\boldsymbol{\mu}_{\boldsymbol{\ell}}^\top [\mathbf{x}_1 \mid \mathbf{x}_2] = 0$, where the matrix $[\mathbf{x}_1 \mid \mathbf{x}_2]$ is the null space of $\boldsymbol{\mu}_{\boldsymbol{\ell}}^\top$. One of these points can be used to determine the sign of $\mathbf{x}_{\boldsymbol{\ell}}^\top \mathtt{C}\mathbf{x}_{\boldsymbol{\ell}}$.

Points and hyperplanes in n dimensions Generalizing the above results from 2 to n dimensions is straightforward. A homogeneous n-dimensional point $\mathbf{X} \in \mathbb{R}^{(n+1)}$ with mean $\boldsymbol{\mu}_{\mathbf{X}}$ and covariance $\Sigma_{\mathbf{X}}$ of rank n is bounded by the homogeneous n-dimensional quadric

$$\mathtt{Q} = \left[\boldsymbol{\mu}_{\mathbf{X}}\boldsymbol{\mu}_{\mathbf{X}}^\top - k^2\Sigma_{\mathbf{X}}\right]^{-1}$$

representing an ellipsoid in n dimensions containing $\boldsymbol{\mu}_{\mathbf{X}}$. By duality, a homogeneous hyperplane $\boldsymbol{\pi}$ is bounded by the dual quadric $\mathtt{Q}^* = \left[\boldsymbol{\mu}_{\boldsymbol{\pi}}\boldsymbol{\mu}_{\boldsymbol{\pi}}^\top - k^2\Sigma_{\boldsymbol{\pi}}\right]^{-1}$ of the same dimension as the hyperplane, where $\boldsymbol{\mu}_{\boldsymbol{\pi}}$ and $\Sigma_{\boldsymbol{\pi}}$ are the mean and covariance, respectively, of the hyperplane. The hyperboloid bounding the uncertainty of the hyperplane is given by the quadric

$$\mathtt{Q} = \boldsymbol{\mu}_{\boldsymbol{\pi}}\boldsymbol{\mu}_{\boldsymbol{\pi}}^\top - k^2\Sigma_{\boldsymbol{\pi}}$$

which is symmetric about the hyperplane.

Two-view geometry

In this section we apply our method to point-to-point mapping under a planar homography and point-to-line mapping under a fundamental matrix. The maximum

likelihood estimate and its covariance are determined from image point correspondences by 2D block adjustment [9] and two-view bundle adjustment [93] for the planar homography and fundamental matrix, respectively. First, points are detected in each of the images using the Förstner operator [26]. For each point in image 1, its initial corresponding point is established by searching for the point in image 2 that has the highest local normalized cross-correlation value. The resulting set of initial point correspondences are used as input to RANSAC [23], which provides both a linear estimate of the model and its set of inlier point correspondences. Lastly, the reprojection error is minimized using a sparse implementation of the Levenberg-Marquardt algorithm [34]. We retrieve the covariance matrix of the parameters after minimization.

For analysis, we select a point $\mathbf{x}$ in image 1 that did not participate in block adjustment. The point $\mathbf{x} = (\tilde{\mathbf{x}}^\top, 1)^\top$ has covariance

$$\Sigma_{\mathbf{x}} = \begin{bmatrix} \tilde{\Sigma}_{\mathbf{x}} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}$$

where the inhomogeneous coordinate $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})^\top$ has assumed covariance $\tilde{\Sigma}_{\mathbf{x}} = \mathtt{I}_{2\times 2}$.

Figure 4.1 shows the results of point-to-point mapping under the estimated planar homography. The mapped point is computed by

$$\mathbf{x}' = \frac{\mathtt{H}\mathbf{x}}{\|\mathtt{H}\mathbf{x}\|}$$

From (4.1), the covariance of $\mathbf{x}'$ is $\Sigma_{\mathbf{x}'} \approx \mathtt{J}_{\mathbf{h}}\Sigma_{\mathbf{h}}\mathtt{J}_{\mathbf{h}}^\top + \mathtt{J}_{\mathbf{x}}\Sigma_{\mathbf{x}}\mathtt{J}_{\mathbf{x}}^\top$, where $\mathtt{J}_{\mathbf{h}}$ and $\mathtt{J}_{\mathbf{x}}$ are computed from (4.4) and (4.5), respectively, and the associated uncertainty ellipse is computed from (4.9). Points that did participate in block adjustment are correlated to the estimated homography. If one of these points were selected, then the cross-covariance $\Sigma_{\mathbf{hx}}$ would be nonzero and the covariance of $\mathbf{x}'$ calculated as

$$\Sigma_{\mathbf{x}'} \approx \begin{bmatrix} \mathtt{J}_{\mathbf{h}} & \mathtt{J}_{\mathbf{x}} \end{bmatrix} \begin{bmatrix} \Sigma_{\mathbf{h}} & \Sigma_{\mathbf{hx}} \\ \Sigma_{\mathbf{xh}} & \Sigma_{\mathbf{x}} \end{bmatrix} \begin{bmatrix} \mathtt{J}_{\mathbf{h}}^\top \\ \mathtt{J}_{\mathbf{x}}^\top \end{bmatrix}$$

Similarly, results for point-to-line mapping under a fundamental matrix are shown in figure 1.8 on page 26. The line $\boldsymbol{\ell}'$ corresponding to the point $\mathbf{x}$ is computed by

$$\boldsymbol{\ell}' = \frac{\mathtt{F}\mathbf{x}}{\|\mathtt{F}\mathbf{x}\|}$$

The covariance of $\boldsymbol{\ell}'$ is $\Sigma_{\boldsymbol{\ell}'} \approx \mathtt{J}_{\mathbf{f}}\Sigma_{\mathbf{f}}\mathtt{J}_{\mathbf{f}}^\top + \mathtt{J}_{\mathbf{x}}\Sigma_{\mathbf{x}}\mathtt{J}_{\mathbf{x}}^\top$ with associated uncertainty hyperbola given by (4.10).

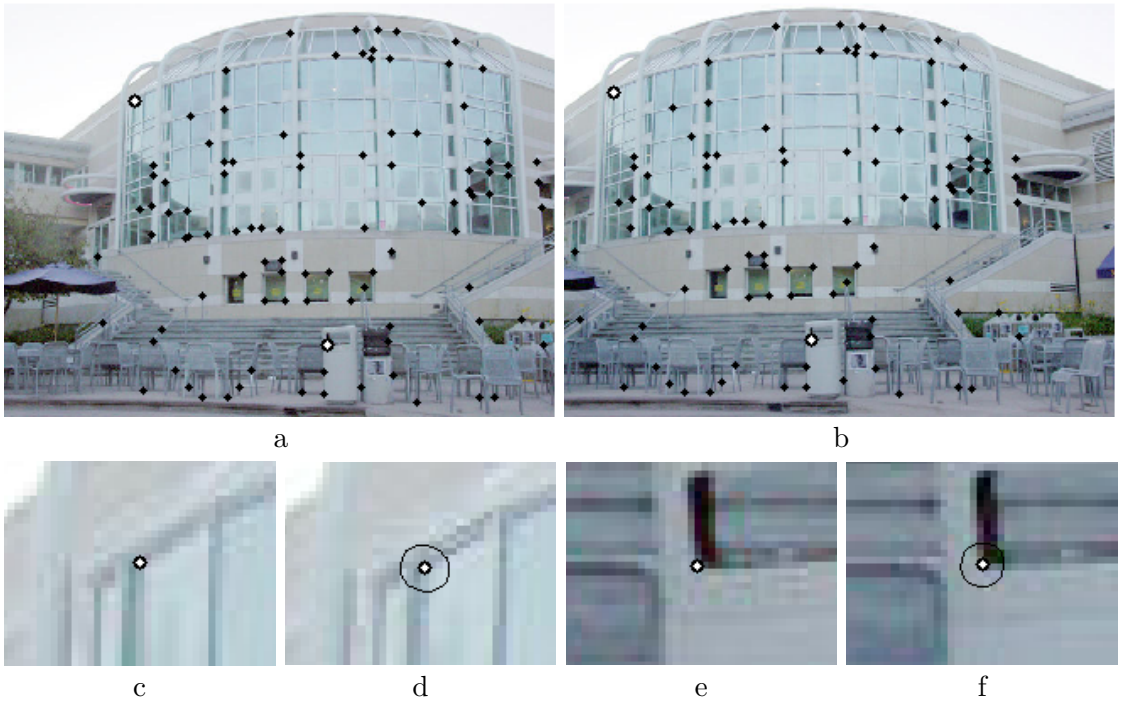


Figure 4.1: Point-to-point mapping under a planar homography. (a) (b) The left and right images acquired from a camera undergoing pure rotation about its center with corresponding points used in 2D block adjustment (in black). There are 97 point correspondences. Two additional points have been selected in the left image (in white) and mapped to the right image (in white). The uncertainty ellipses associated with the mapped points are contained in the right image (in black). The ellipses correspond to a probability of 99%. (c) The left image zoomed in on the first selected point. (d) The right image zoomed in on the corresponding first mapped point. (e) The left image zoomed in on the second selected point. (f) The right image zoomed in on the corresponding second mapped point. Note that the eccentricity of the ellipse associated with the first point is slightly greater than that of the ellipse associated with the second point. This is because the second point is surrounded by points used in block adjustment, while the first point is not.

Mosaic construction from video

This section describes use of our approach in the application of video mosaicing. More specifically, we apply our approach to the special case of the video looping back on itself, i.e., the sensor returns to image a region of the scene that it imaged at a previous time. We seek to determine the search region in the previously acquired frames that spatially overlap with the looped back frames but are not temporal neighbors with these frames. Mosaic construction from video is performed in a sequential manner as follows.

For each video frame, features are detected using the method described in [88]. This method detects windows of bidirectional texturedness, which are good features to track in video. Nonmaxima suppression is applied to detected features to limit their number. A pyramidal implementation of Lucas-Kanade [52] determines the translation of each feature from the current frame to the previous one. Just as with two-view estimation, RANSAC is applied to the inter-frame correspondences, and the reprojection error is minimized using a sparse implementation of the Levenberg-Marquardt algorithm and the covariance matrix is retrieved. The homographies are accumulated such that the current frame n is mapped back to frame 1 of the video by

$$\mathtt{H}_{n,1} = \frac{\mathtt{H}_{n-1,1}\mathtt{H}_{n,n-1}}{\|\mathtt{H}_{n-1,1}\mathtt{H}_{n,n-1}\|} = \frac{\mathtt{AB}}{\|\mathtt{AB}\|}$$

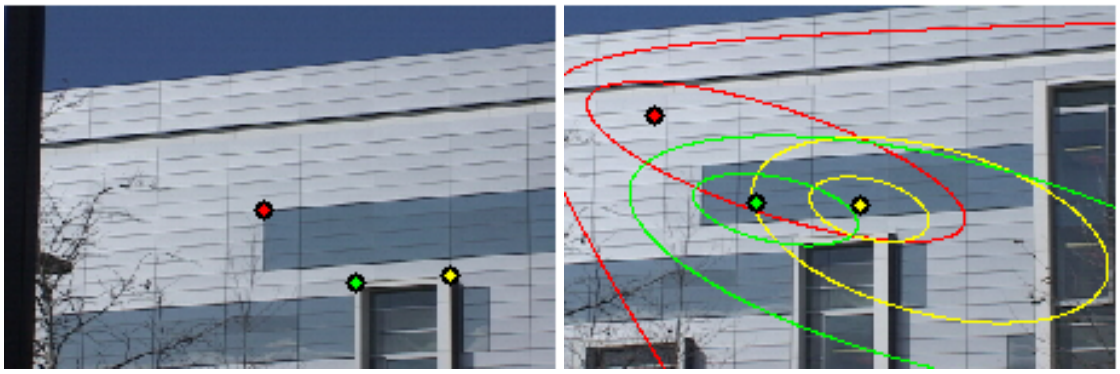
and the covariance of $\mathtt{H}_{n,1}$ is approximately $\mathtt{J}_{\mathbf{a}}\Sigma_{\mathbf{a}}\mathtt{J}_{\mathbf{a}}^{\top} + \mathtt{J}_{\mathbf{b}}\Sigma_{\mathbf{b}}\mathtt{J}_{\mathbf{b}}^{\top}$.

As the homographies between successive frames $\mathtt{H}_{n,n-1}$ are accumulated, so are their uncertainties. It is expected that the uncertainty of $\mathtt{H}_{n,1}$ will increase with n, i.e., looped back frames will not align with previous frames containing images of the same region of the scene and this misalignment will increase as the time between these frames increases. Figure 4.2 illustrates the results of this approach.

4.1.2 Feature matching

When it is determined that multiple cameras are imaging the same region of a scene, the SIFT features detected in each of the images are robustly matched using the approach described in this section. The feature matching process first establishes a set of putative correspondences between SIFT features that have been detected in each of the images acquired by the cameras. Putative correspondences are computed using a combination of guided matching (described above) and the comparison of SIFT descriptors. RANSAC is then applied to the set of putative feature correspondences to

a



b c

Figure 4.2: Mosaic construction from video. (a) A planar mosaic sequentially constructed from a video containing 706 frames acquired from a camera undergoing pure rotation about its center. The video begins at the upper left corner of the face of the building and moves clockwise around the border of the face returning to the upper left corner. Note that the last frame is not aligned with the first frame due to uncertainty accumulation. (b) (c) The last and first frames (images) of the video. Three points (in red, yellow, and green) have been selected in the last image and mapped to the first image. The uncertainty ellipses associated with the mapped points are contained in the first image (also in red, yellow, and green). The ellipses correspond to probabilities of 50% and 99% for each mapped point. In this case, the corresponding points in the first image are contained in the ellipses corresponding to a probability of 50%.

determine the set of inlier correspondences. Presently, the matching process is tailored to work on pairs on images. In the case of three or more cameras imaging the same region of the scene, all possible image pairs are processed and the results merged.

Putative correspondences

To determine the search region for putative feature correspondences between pairs of images acquired from two different cameras, covariance propagation through the essential matrix is used. The essential matrix is a specialization of the fundamental matrix. Both the essential matrix $\mathtt{E}$ and the fundamental matrix $\mathtt{F}$ are epipolar transformations that map points in an image acquired by one camera to lines containing the corresponding point in an image acquired by a second camera, where there is a nonzero distance between the centers of the two cameras. However, unlike the fundamental matrix, which maps points (in pixel coordinates) in the first image to lines (in pixel coordinates) in the second image, the essential matrix maps points in normalized coordinates from the first image to lines in normalized coordinates in the second image.

In order to work in the space of normalized coordinates, the matching process converts the detected SIFT features from image coordinates to normalized coordinates with uncertainty propagation as described in the section on inverse mapping (page 36). Additionally, the current estimate of the Kalman filter state vector of each camera $\mathbf{x} = (\boldsymbol{\omega}^\top, \dot{\boldsymbol{\omega}}^\top, \widetilde{\mathbf{C}}^\top, \dot{\widetilde{\mathbf{C}}}^\top)^\top$ is mapped to a vector $(\boldsymbol{\omega}^\top, \mathbf{t}^\top)^\top$ containing the parameters of the normalized camera $\hat{\mathtt{P}} = [\mathtt{R} \mid \mathbf{t}] = [\exp(\boldsymbol{\omega}) \mid \mathbf{t}]$. The covariance matrix $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)}$ associated with the vector $(\boldsymbol{\omega}^\top, \mathbf{t}^\top)^\top$ is calculated by $\Sigma_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)} \approx \mathtt{J}_\mathbf{x} \Sigma_\mathbf{x} \mathtt{J}_\mathbf{x}^\top$, where $\mathbf{t} = -\exp(\boldsymbol{\omega})\widetilde{\mathbf{C}}$ and the Jacobian matrix $\mathtt{J}_\mathbf{x}$ is given by

$$\mathtt{J}_\mathbf{x} = \frac{\partial(\boldsymbol{\omega}^\top, \mathbf{t}^\top)}{\partial \mathbf{x}} = \begin{bmatrix} \mathtt{I} & \mathtt{0} & \mathtt{0} & \mathtt{0} \\ \frac{\partial \mathbf{t}}{\boldsymbol{\omega}} & \mathtt{0} & \frac{\partial \mathbf{t}}{\partial \widetilde{\mathbf{C}}} & \mathtt{0} \end{bmatrix}$$

where $\partial \mathbf{t}/\boldsymbol{\omega}$ and $\partial \mathbf{t}/\partial \widetilde{\mathbf{C}}$ are derived as follows. Let

$$\begin{aligned} \mathbf{t} &= -\exp(\boldsymbol{\omega})\widetilde{\mathbf{C}} \\ \mathbf{t} &= -\mathtt{R}\widetilde{\mathbf{C}} \\ \mathbf{t} &= \mathtt{A}\widetilde{\mathbf{C}} \end{aligned}$$

where $\mathtt{R} = \exp(\boldsymbol{\omega})$ and $\mathtt{A} = -\mathtt{R}$ with $\partial \mathbf{r}/\partial \boldsymbol{\omega}$ given by (2.9), $\partial \mathbf{a}/\partial \mathbf{r} = -\mathtt{I}$, $\partial \mathbf{t}/\partial \mathbf{a}$ given by

(A.12), $\partial\mathbf{t}/\partial\widetilde{\mathbf{C}}$ given by (A.13), and

$$\frac{\partial \mathbf{t}}{\partial \boldsymbol{\omega}} = \frac{\partial \mathbf{t}}{\partial \mathbf{a}} \frac{\partial \mathbf{a}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\omega}}$$

For use in guided matching between images acquired by every combination of camera pairs, the essential matrix $\mathtt{E}$ from a given pair of general normalized cameras $\hat{\mathtt{P}} = [\mathtt{R} \mid \mathbf{t}]$ and $\hat{\mathtt{P}}' = [\mathtt{R}' \mid \mathbf{t}']$ is given by

$$\mathtt{E} = \frac{[\mathbf{t}' - \mathtt{R}'\mathtt{R}^\top\mathbf{t}]_\times \mathtt{R}'\mathtt{R}^\top}{\|\mathbf{t}' - \mathtt{R}'\mathtt{R}^\top\mathbf{t}]_\times \mathtt{R}'\mathtt{R}^\top\|}$$

where $\mathtt{R} = \exp(\boldsymbol{\omega})$ and $\mathtt{R}' = \exp(\boldsymbol{\omega}')$. The covariance matrix $\boldsymbol{\Sigma}_\mathbf{e}$ associated with $\mathtt{E}$ is calculated as $\boldsymbol{\Sigma}_\mathbf{e} \approx \mathtt{J}\boldsymbol{\Sigma}_{(\boldsymbol{\omega}^\top, \mathbf{t}^\top)}\mathtt{J}^\top + \mathtt{J}'\boldsymbol{\Sigma}_{(\boldsymbol{\omega}'^\top, \mathbf{t}'^\top)}\mathtt{J}'^\top$, where

$$\mathtt{J} = \begin{bmatrix} \frac{\partial \mathbf{e}}{\partial \boldsymbol{\omega}} & \frac{\partial \mathbf{e}}{\partial \mathbf{t}} \end{bmatrix} \quad \text{and} \quad \mathtt{J}' = \begin{bmatrix} \frac{\partial \mathbf{e}}{\partial \boldsymbol{\omega}'} & \frac{\partial \mathbf{e}}{\partial \mathbf{t}'} \end{bmatrix}$$

where $\partial\mathbf{e}/\partial\boldsymbol{\omega}$, $\partial\mathbf{e}/\partial\mathbf{t}$, $\partial\mathbf{e}/\partial\boldsymbol{\omega}'$, and $\partial\mathbf{e}/\partial\mathbf{t}'$ are derived as follows. For clarity let, $\mathtt{A} = \mathtt{R}^\top$, $\mathtt{N} = \mathtt{R}'\mathtt{A}$, $\mathbf{b} = \mathtt{N}\mathbf{t}$, $\mathbf{c} = \mathbf{t}' - \mathbf{b}$, $\mathtt{D} = [\mathbf{c}]_\times$, $\mathtt{M} = \mathtt{N}$, $\mathtt{G} = \mathtt{D}\mathtt{M}$, and $\mathtt{E} = \mathtt{G}/\|\mathtt{G}\|$.

$$\frac{\partial \mathbf{e}}{\partial \boldsymbol{\omega}} = \frac{\partial \mathbf{e}}{\partial \mathbf{n}} \frac{\partial \mathbf{n}}{\partial \mathbf{a}} \frac{\partial \mathbf{a}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\omega}} \quad \text{and} \quad \frac{\partial \mathbf{e}}{\partial \mathbf{t}} = \frac{\partial \mathbf{e}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{d}} \frac{\partial \mathbf{d}}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \mathbf{t}}$$
$$\frac{\partial \mathbf{e}}{\partial \boldsymbol{\omega}'} = \frac{\partial \mathbf{e}}{\partial \mathbf{n}} \frac{\partial \mathbf{n}}{\partial \mathbf{r}'} \frac{\partial \mathbf{r}'}{\partial \boldsymbol{\omega}'} \quad \text{and} \quad \frac{\partial \mathbf{e}}{\partial \mathbf{t}'} = \frac{\partial \mathbf{e}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{d}} \frac{\partial \mathbf{d}}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial \mathbf{t}'}$$

where $\partial\mathbf{m}/\partial\mathbf{n} = \mathtt{I}$; $\partial\mathbf{r}/\partial\boldsymbol{\omega}$ and $\partial\mathbf{r}'/\partial\boldsymbol{\omega}'$ are given by (2.9); $\partial\mathbf{a}/\partial\mathbf{r}$, $\partial\mathbf{n}/\partial\mathbf{r}'$, $\partial\mathbf{n}/\partial\mathbf{a}$, $\partial\mathbf{b}/\partial\mathbf{n}$, $\partial\mathbf{b}/\partial\mathbf{t}$, $\partial\mathbf{c}/\partial\mathbf{t}'$, $\partial\mathbf{c}/\partial\mathbf{b}$, $\partial\mathbf{d}/\partial\mathbf{c}$, $\partial\mathbf{g}/\partial\mathbf{d}$, $\partial\mathbf{g}/\partial\mathbf{m}$, and $\partial\mathbf{e}/\partial\mathbf{g}$ are given by equations found in section A.1 on page 78; and

$$\frac{\partial \mathbf{e}}{\partial \mathbf{n}} = \frac{\partial \mathbf{e}}{\partial \mathbf{g}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{d}} \frac{\partial \mathbf{d}}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \mathbf{n}} + \frac{\partial \mathbf{g}}{\partial \mathbf{m}} \frac{\partial \mathbf{m}}{\partial \mathbf{n}} \right)$$

The resulting essential matrix $\mathtt{E}$ and covariance $\boldsymbol{\Sigma}_\mathbf{e}$ are used to calculate search regions in image 2. A search region in image 2 corresponding to a detected SIFT feature in image 1 is determined by the mapping

$$\hat{\boldsymbol{\ell}}' = \frac{\mathtt{E}\hat{\mathbf{x}}}{\|\mathtt{E}\hat{\mathbf{x}}\|}$$

where $\hat{\mathbf{x}}$ is the point in normalized coordinates in image 1 and $\hat{\boldsymbol{\ell}}'$ is the line in normalized coordinates in image 2 with covariance $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\ell}}'} \approx \mathtt{J}_\mathbf{e}\boldsymbol{\Sigma}_\mathbf{e}\mathtt{J}_\mathbf{e}^\top + \mathtt{J}_{\hat{\mathbf{x}}}\boldsymbol{\Sigma}_{\hat{\mathbf{x}}}\mathtt{J}_{\hat{\mathbf{x}}}^\top$ with associated uncertainty hyperbola given by (4.10). As described in section 4.1.1, the hyperbola in image 2 bounds the region (at the specified probability) that contains the SIFT feature corresponding to $\hat{\mathbf{x}}$ in image 1. SIFT features detected in image 2 that are contained within

the branches of the hyperbola meet the geometric criteria for potentially corresponding to $\hat{\mathbf{x}}$.

Next, the feature descriptors are compared to determine the similarity between features within the search region of image 2 and the feature in image 1 at $\hat{\mathbf{x}}$. The matching process calculates how similar the potential corresponding features are as well as how unique the potential match is. For a detected SIFT feature in image 1, the matching process measures the Euclidean distance between its associated SIFT descriptor vector and all descriptor vectors contained in its corresponding search region in image 2, storing the distances to its nearest and second nearest neighbors, i.e., the smallest and second smallest Euclidean distances. The ratio of the smallest distance to the second smallest distance is a measure of how ambiguous the match is [51]. The lower the ratio, the less ambiguous the match. Thresholding on this ratio is an effective method for removing ambiguous matches. In this work, a threshold of 0.8 consistently resulted in sets of reliable matches.

Last, as described in section 3.1.1 on page 52, RANSAC is applied to the resulting set of putative correspondences to determine the subset of correspondences that are consistent with the essential matrix. Figure 1.7 on page 23 shows example results of matching features across images acquired from different cameras using the procedure described in this section.

4.2 Joint estimation of geoposition and orientation

Prior to two or more cameras imaging the same region of the scene and establishing feature correspondences between their images, independent processes have been estimating the position and orientation of each camera over time using methods described in the previous chapter. Matching features between images acquired by different cameras introduces the sharing of information across the cameras. There are multiple techniques for combining this additional information in order to improve the estimates of the positions and orientations of the cameras. These approaches range from a single Kalman filter with a state vector containing all of the parameters for all of the cameras to, for example, a decentralized data fusion framework [67]. Most of these approaches have been developed to mitigate issues that arise when the number of cameras significantly increases, for example, from tens of cameras to tens of thousands. Those techniques

that do scale to a large number of cameras must often sacrifice some information for the ability to scale. The approach developed in this work falls into this category.

The method used in this work is a hybrid one. Each camera continues to independently estimate its position and orientation as described in the previous chapter. However, when two or more cameras image the same region of a scene and feature correspondences are established, a separate, independent process will simultaneously estimate the position and orientation of these cameras given their current position and orientation estimates, and the set of feature correspondences between their images—a process called bundle adjustment [93]. The results of the bundle adjustment process are then input to each of the Kalman filters as simply another measurement and position and orientation. After the measurement update, the Kalman filters return to independent processing. The information that is lost by using this approach is the cross-camera covariance information resulting from bundle adjustment.

Bundle adjustment [93] can reliably estimate positions and orientations of hundreds of cameras simultaneously. In the framework described above, the position and orientation of each camera is estimated independently, enabling it to be performed on the camera, if desired. Bundle adjustment need only be performed for subsets of cameras that are imaging the same region of a scene at the same time. Under typical conditions, multiple bundle adjustment processes will be executing, each adjusting perhaps tens of cameras, which is easily handled. With this in mind, the loss of cross-camera covariance information is considered an acceptable loss. The advantage of bundle adjustment is, through the use of cross-camera image feature correspondences, it allows the cameras to transfer their accuracy to each other by jointly estimating the position and orientation of the cameras.

A sparse implementation of the Levenberg-Marquardt algorithm [34] is used to perform bundle adjustment, allowing computationally efficient adjustment the geoposition and orientation of m cameras as follows. For clarity, the hat notation is removed from the normalized image coordinates in the measurement vector $\mathbf{X}$ so that they are not confused with the estimate of the measurement vector $\widehat{\mathbf{X}}$. The initial estimate of the parameter vector is

$$\widehat{\mathbf{P}} = (\boldsymbol{\omega}^{(1)\top}, \widetilde{\mathbf{C}}^{(1)\top}, \ldots, \boldsymbol{\omega}^{(m)\top}, \widetilde{\mathbf{C}}^{(m)\top}, \widetilde{\mathbf{X}}_1^{\top}, \ldots, \widetilde{\mathbf{X}}_n^{\top})^{\top}$$

where the jth camera rotation $\boldsymbol{\omega}^{(j)}$ and center $\widetilde{\mathbf{C}}^{(j)}$ are initialized to the values in the jth Kalman filter state vector for all m cameras imaging the scene, and the 3D points $\widetilde{\mathbf{X}}_i \; \forall \; i$

are initialized by triangulation using the DLT algorithm. The measurement vector $\mathbf{X}$ is given by

$$\mathbf{X} = (\boldsymbol{\omega}^{(1)\top}, \widetilde{\mathbf{C}}^{(1)\top}, \tilde{\mathbf{x}}_1^{(1)\top}, \ldots, \tilde{\mathbf{x}}_n^{(1)\top}, \ldots, \boldsymbol{\omega}^{(m)\top}, \widetilde{\mathbf{C}}^{(m)\top}, \tilde{\mathbf{x}}_1^{(m)\top}, \ldots, \tilde{\mathbf{x}}_n^{(m)\top})^\top$$

where $\tilde{\mathbf{x}}_i^{(j)}$ is the ith RANSAC inlier point in normalized coordinates in the jth camera. Notice that the above measurement vector also includes the current Kalman filter state estimates of the rotations and translations of the cameras. Inclusion of the rotations and translations in the measurement vector prevents their counterparts in the parameter vector from being adjusted outside of the uncertainty bounds of the current state estimate.

After bundle adjustment, the resulting rotations and translations are extracted from the final estimate of the parameter vector and their covariances retrieved. Measurement updates of rotation and translation are issued to each Kalman filter associated with an adjusted camera. The result is decreased relative error between the cameras, resulting in more precise estimates of the geoposition and orientation of the cameras.

Acknowledgment Section 4.1.1 of this chapter is based on the paper "Covariance Propagation for Guided Matching" by B. Ochoa and S. Belongie [72]. I was the primary investigator and author of this paper.

A

Appendix

A.1 Partial derivatives of matrix operations

Propagation of covariance is extensively performed throughout the dissertation. Covariance propagation through matrix operations requires calculation of the Jacobian matrices associated with the operations. This appendix list analytical expressions of Jacobian matrices for several common matrix operations.

Notation

The following notation is used in this appendix:

- If a capital letter is used to denote a matrix, then the vector denoted by the corresponding lower case letter is composed of the entries of the matrix by

$$\mathtt{A} \in \mathbb{R}^{m \times n} \Leftrightarrow \mathtt{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix}, \mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{pmatrix} \in \mathbb{R}^{mn}$$

  where $\mathbf{a}_i^\top \in \mathbb{R}^n$ is the ith row of $\mathtt{A}$ (i.e., $\mathbf{a} = \mathrm{vec}(\mathtt{A}^\top)$).

- $\otimes$ denotes the Kronecker product

- The set of vectors $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ is the standard basis in the vector space $\mathbb{R}^n$ (e.g., $\mathbf{e}_2 = (0, 1, 0, \ldots, 0)^\top$).

Matrix operations and their associated Jacobian matrices

matrix norm Let $\mathtt{A} \in \mathbb{R}^{m \times n}$. For the equation $s = \|A\| \in \mathbb{R}$,

$$\frac{\mathrm{d}s}{\partial \mathbf{a}} = \frac{1}{s}\mathbf{a}^\top \tag{A.1}$$

matrix trace Let $\mathtt{A} \in \mathbb{R}^{n \times n}$. For the equation $s = \mathrm{trace}(\mathtt{A}) \in \mathbb{R}$,

$$\frac{\mathrm{d}s}{\partial \mathbf{a}} = (\mathbf{e}_1^\top \mid \mathbf{e}_2^\top \mid \ \ldots \ \mid \mathbf{e}_n^\top) \tag{A.2}$$

matrix transpose Let $\mathtt{A} \in \mathbb{R}^{m \times n}$. For the equation $\mathtt{B} = \mathtt{A}^\top \in \mathbb{R}^{n \times m}$,

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \begin{bmatrix} \mathtt{I}_{m \times m} \otimes \mathbf{e}_1^\top \\ \mathtt{I}_{m \times m} \otimes \mathbf{e}_2^\top \\ \vdots \\ \mathtt{I}_{m \times m} \otimes \mathbf{e}_n^\top \end{bmatrix} \tag{A.3}$$

scalar-matrix multiplication Assume $s \in \mathbb{R}$, $\mathtt{A} \in \mathbb{R}^{m \times n}$, and $\mathtt{B} \in \mathbb{R}^{m \times n}$. For the equation $\mathtt{B} = s\mathtt{A}$,

$$\frac{\partial \mathbf{b}}{\mathrm{d}s} = \mathbf{a} \tag{A.4}$$

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = s\mathtt{I} \tag{A.5}$$

matrix-scalar division Assume $s \in \mathbb{R}$, $\mathtt{A} \in \mathbb{R}^{m \times n}$, and $\mathtt{B} \in \mathbb{R}^{m \times n}$. For the equation $\mathtt{B} = \mathtt{A}/s$,

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \mathtt{I}/s \tag{A.6}$$

$$\frac{\partial \mathbf{b}}{\mathrm{d}s} = -\mathbf{a}/s^2 \tag{A.7}$$

matrix-matrix addition Assume $\mathtt{A} \in \mathbb{R}^{m \times n}$, $\mathtt{B} \in \mathbb{R}^{m \times n}$, and $\mathtt{C} \in \mathbb{R}^{m \times n}$. For the equation $\mathtt{C} = \mathtt{A} + \mathtt{B}$,

$$\frac{\partial \mathbf{c}}{\partial \mathbf{a}} = \mathtt{I} \tag{A.8}$$

$$\frac{\partial \mathbf{c}}{\partial \mathbf{b}} = \mathtt{I} \tag{A.9}$$

matrix-matrix subtraction Assume $\mathtt{A} \in \mathbb{R}^{m \times n}$, $\mathtt{B} \in \mathbb{R}^{m \times n}$, and $\mathtt{C} \in \mathbb{R}^{m \times n}$. For the equation $\mathtt{C} = \mathtt{A} - \mathtt{B}$,

$$\frac{\partial \mathbf{c}}{\partial \mathbf{a}} = \mathtt{I} \tag{A.10}$$

$$\frac{\partial \mathbf{c}}{\partial \mathbf{b}} = -\mathtt{I} \tag{A.11}$$

matrix-matrix multiplication Assume $\mathtt{A} \in \mathbb{R}^{m \times p}$, $\mathtt{B} \in \mathbb{R}^{p \times n}$, and $\mathtt{C} \in \mathbb{R}^{m \times n}$. For the equation $\mathtt{C} = \mathtt{AB}$,

$$\frac{\partial \mathbf{c}}{\partial \mathbf{a}} = \mathtt{I}_{m \times m} \otimes \mathtt{B}^\top \tag{A.12}$$

$$\frac{\partial \mathbf{c}}{\partial \mathbf{b}} = \mathtt{A} \otimes \mathtt{I}_{n \times n} \tag{A.13}$$

3 × 3 skew-symmetric matrix corresponding to 3-vector Let $\mathbf{v} = (v_1, v_2, v_3)^\top$. For the equation

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{A.14}$$

the Jacobian matrix

$$\frac{\partial \operatorname{vec}(([\mathbf{v}]_\times)^\top)}{\partial \mathbf{v}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.15}$$

matrix unitization Let $\mathtt{A} \in \mathbb{R}^{m \times n}$. For clarity, let $\mathtt{B} = \mathtt{A}$. For the equation

$$\mathtt{C} = \frac{\mathtt{A}}{\|\mathtt{A}\|} \in \mathbb{R}^{m \times n} \tag{A.16}$$

$$\mathtt{C} = \frac{\mathtt{B}}{\|\mathtt{A}\|} \tag{A.17}$$

the Jacobian matrix

$$\frac{\partial \mathbf{c}}{\partial \mathbf{a}} = \frac{\partial \mathbf{c}}{\partial \|\mathtt{A}\|}\frac{\partial \|\mathtt{A}\|}{\partial \mathbf{a}} + \frac{\partial \mathbf{c}}{\partial \mathbf{b}}\frac{\partial \mathbf{b}}{\partial \mathbf{a}} \tag{A.18}$$

where $\partial \mathbf{c}/\partial \|\mathtt{A}\|$ is given by (A.7), $\partial \|\mathtt{A}\|/\partial \mathbf{a}$ is given by (A.1), $\partial \mathbf{c}/\partial \mathbf{b}$ is given by (A.6), and $\partial \mathbf{b}/\partial \mathbf{a} = \mathtt{I}$.

Bibliography

[1] A9 Maps. `http://maps.a9.com/`. Discontinued on September 29, 2006.

[2] Advanced Orientation Systems, Inc. *EZ-COMPASS-3 Application Manual*.

[3] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proceedings of the European Conference on Computer Vision*, pages 683–695, 1996.

[4] J.-Y. Bouguet. A camera calibration toolbox for MATLAB®. `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`.

[5] R. Bowen. Global positioning system operational control system accuracies. *Navigation*, 32(2), 1985.

[6] D. C. Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.

[7] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[8] R. S. Bucy. Nonlinear filtering theory. *IEEE Transactions on Automatic Control*, 10(2):198, 1965.

[9] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 885–891, 1998.

[10] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535, 2002.

[11] T. A. Clarke and J. G. Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91):51–66, 1998.

[12] A. E. Conrady. Decentered lens-systems. *Monthly Notices of the Royal Astronomical Society*, 79(3):384–390, 1919.

[13] E. M. Copps. An aspect of the role of the clock in a GPS receiver. *Navigation*, 31(3), 1984.

[14] A. Criminisi. A plane measuring device. *Image and Vision Computing*, 40(2):625–634, 1999.

[15] G. Csurka, C. Zeller, Z. Zhang, and O. D. Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 68(1):18–36, 1997.

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

[17] Defense Mapping Agency. *The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS)*, 1989. DMA TM 8358.2.

[18] Defense Mapping Agency. *Datums, Ellipsoids, Grids, and Grid Reference Systems*, 1990. DMA TM 8358.1.

[19] Department of Defense. *Department of Defense Glossary of Mapping, Charting, and Geodetic Terms*, 1994. MIL-HDBK-850.

[20] Department of Defense. *Department of Defense World Geodetic System (WGS)*, 1994. MIL-STD-2401.

[21] Department of Defense/Intelligence Community/National System for Geospatial Intelligence. *Motion Imagery Standards Profile*, 2006. Version 4.0.

[22] D. D. Diel, P. DeBitetto, and S. Teller. Epipolar constraints for vision-aided inertial navigation. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 221–228, 2005.

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

[24] A. W. Fitzgibbon and A. Zisserman. Automatic camera tracking. In *Video Registration*, pages 18–35. Kluwer, 2003.

[25] W. Förstner. Uncertainty and projective geometry. In E. B. Corrochano, editor, *Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics*, pages 493–534. Springer-Verlag, 2005.

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

[27] J. G. Fryer and D. C. Brown. Lens distortion for close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 52(1):51–58, 1986.

[28] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.

[29] GeoSpatial Experts. *GPS-Photo Link User Manual*.

[30] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

[31] A. Graham. *Kronecker Products and Matrix Calculus with Applications*. Ellis Horwood Ltd., 1981.

[32] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[33] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[34] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Workshop on Applications of Invariance in Computer Vision*, pages 237–256, 1994.

[35] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[36] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.

[37] Holux. *Holux GPSlim236 Wireless Bluetooth GPS Receiver Specification*, 2005.

[38] S. Hsu. Geocoded terrestrial mosaics using pose sensors and video registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 834–841, 2001.

[39] ITU Telecommunication Standardization Sector. *Video Codec for Autiovisual Services at $p \times 64$ kbit/s*, 1993. ITU-T Recommendation H.261.

[40] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the European Conference on Computer Vision*, pages 288–241, 2004.

[41] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, pages 35–45, 1960.

[42] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996.

[43] T. Kindberg, M. Spasojevic, R. Fleck, and A. Sellen. The ubiquitous camera: An in-depth study of camera phone use. *IEEE Pervasive Computing*, 4(2):42–50, 2005.

[44] R. Kumar, S. Samarasekera, S. Hsu, and K. Hanna. Registration of highly-oblique and zoomed in aerial video to reference imagery. In *Proceedings of the International Conference on Pattern Recognition*, pages 303–307, 2000.

[45] F. G. Lemoine et al. The development of the joint NASA GSFC and NIMA geopotential model EGM96. Technical report, National Aeronautics and Space Administration, 1998. NASA/TP-1998-206861.

[46] K. Levenberg. A method for the solution of certain problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168, 1944.

[47] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[48] D. Lowe. Demo software: SIFT keypoint detector. `http://www.cs.ubc.ca/~lowe/keypoints/`.

[49] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, 1999.

[50] D. G. Lowe. Local feature view clustering for 3D object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–688, 2001.

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

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

[53] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2004.

[54] E. Marianovsky. Personal correspondence to Ben Ochoa. EZ-COMPASS measurement uncertainties.

[55] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[56] E. H. Martin. GPS user eqipment error models, 1977. Institute of Navigation Annual Meeting.

[57] P. Massatt and K. Rudnick. Geometric formulas for dilution of precision calculations. *Navigation*, 37(4), 1990.

[58] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[59] C. McGlone, editor. *Manual of Photogrammetry*. ASPRS, fifth edition, 2004.

[60] Microsoft Corporation. *Microsoft LifeCam VX-6000 Technical Data Sheet*, 2006.

[61] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the European Conference on Computer Vision*, pages 128–142, 2002.

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

[63] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–71, 2005.

[64] R. J. Milliken and C. J. Zoller. Principle of operation of NAVSTAR and system characteristics. *Navigation*, 25(2), 1978.

[65] Motorola. *Motorola V325 User Manual*, 2005. 6809494A09-A.

[66] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulaion*. CRC Press, 1993.

[67] E. M. Nebot, M. Bozorg, and H. F. Durrant-Whyte. Decentralized architecture for asynchronous sensors. *Autonomous Robots*, 6(2):147–164, 1999.

[68] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659, 2004.

[69] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

[70] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, second edition, 2006.

[71] B. Ochoa. Personal correspondence to Richard Hartley and Andrew Zisserman. Book corrections, June 2006.

[72] B. Ochoa and S. Belongie. Covariance propagation for guided matching. Workshop on Statistical Methods in Multi-Image and Video Processing 2006.

[73] T. Papadopoulo and M. I. A. Lourakis. Estimating the Jacobian of the singular value decomposition: Theory and applications. In *Proceedings of the European Conference on Computer Vision*, pages 554–570, 2000.

[74] B. W. Parkinson. GPS error analysis. In B. W. Parkinson and J. J. Spilker, Jr, editors, *Global Positioning System: Theory and Applications*, volume 1. American Institute of Aeronautics and Astronautics, 1996.

[75] B. W. Parkinson and J. J. Spilker, Jr, editors. *Global Positioning System: Theory and Applications*, volume 1. American Institute of Aeronautics and Astronautics, 1996.

[76] B. W. Parkinson and J. J. Spilker, Jr, editors. *Global Positioning System: Theory and Applications*, volume 2. American Institute of Aeronautics and Astronautics, 1996.

[77] M. Pollefeys and D. Nistér. Urban 3d modelling from video. `http://cs.unc.edu/Research/urbanscape/`.

[78] R. H. Rapp. *Geometric Geodesy*, volume 1. Ohio State University, Department of Geodetic Sciences, Columbus, Ohio, 1984.

[79] R. H. Rapp. *Geometric Geodesy*, volume 2. Ohio State University, Department of Geodetic Sciences, Columbus, Ohio, 1987.

[80] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 2003.

[81] P. D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, 18(1):97–108, 1982.

[82] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proceedings of the European Conference on Computer Vision*, pages 414–431, 2002.

[83] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[84] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[85] S. M. Seitz and P. Anandan. Implicit representation and scene reconstruction from probability density functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 28–34, 1999.

[86] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[87] M. Shah and R. Kumar, editors. *Video Registration*. Kluwer, 2003.

[88] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[89] J. P. Snyder. *Map Projections – A Working Manual*. Number 1395 in U.S. Geological Survey professional paper. U.S. Government Printing Office, 1987.

[90] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3), 1996.

[91] P. F. Sturm and S. J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1432–1437, 1999.

[92] B. Triggs. Joint feature distributions for image correspondence. In *Proceedings of the International Conference on Computer Vision*, pages 201–108, 2001.

[93] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice*. Springer-Verlag, 2000.

[94] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 4(RA-3):323– 344, 1987.

[95] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[96] T. Tuytelaars, L. Van Gool, L. D'haene, and R. Koch. Matching of affinely invariant regions for visual servoing. In *Proceedings of the International Conference on Robotics and Automation*, pages 1601–1606, 1999.

[97] U.S. Army Corps of Engineers. *Handbook for Transformation of Datums, Projections, Grids and Common Coordinate Systems*, 1996. TEC-SR-7.

[98] R. P. Wildes, D. J. Hirvonen, S. C. Hsu, R. Kumar, W. B. Lehman, B. Matei, and W.-Y. Zhao. Video georegistration: Algorithm and quantitative evaluation. In *Proceedings of the International Conference on Computer Vision*, pages 343–350, 2001.

[99] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998.

[100] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the International Conference on Computer Vision*, pages 666–673, 1999.