Color detection and three-dimensional reconstruction for hand pose estimation as part of an HCI system

Maria Beatriz Teruel
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation
https://digitalscholarship.unlv.edu/rtds/1987

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
COLOR DETECTION AND 3D RECONSTRUCTION FOR HAND POSE

ESTIMATION AS PART OF AN HCI SYSTEM

by

Maria Beatriz Teruel

Bachelor of Science in Mathematical Sciences
National Southern University
Bahia Blanca, Argentina
1999

Master of Science in Mathematical Sciences
University of Nevada, Las Vegas
2003

A thesis submitted in partial fulfillment
of the requirement for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
May 2006

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The Thesis prepared by

MARIA B. TERUEL

Entitled

COLOR DETECTION AND 3D RECONSTRUCTION FOR HAND POSE ESTIMATION AS PART OF AN HCI SYSTEM.

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College
ABSTRACT

Color Detection and 3D Reconstruction for Hand Pose Estimation as Part of an HCI System

by

Maria Beatriz Teruel

Dr. Evangelos A. Yfantis, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

The main focus of this thesis is to give a detailed description of techniques and methodology implemented for feature extraction and three-dimensional reconstruction. The algorithms were developed as part of a project whose main object is the graphical three-dimensional modeling of the gestures of a human hand. The project itself comprises several steps, namely, camera calibration, hand feature extraction, three-dimensional reconstruction, and computer graphics modeling of the hand. Experimental results of the feature extraction and three-dimensional reconstruction will also be presented.
TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... iii

LIST OF FIGURES ........................................................................................................... vi

ACKNOWLEDGMENTS ............................................................................................... viii

CHAPTER 1  INTRODUCTION ....................................................................................... 1

CHAPTER 2  PROJECT DESCRIPTION .......................................................................... 3

CHAPTER 3  BACKGROUND INFORMATION .......................................................... 8

3.1 Camera Calibration .......................................................................................... 8

3.2 Epipolar Geometry ......................................................................................... 12

CHAPTER 4  FEATURE EXTRACTION ......................................................................... 20

4.1 Marker Detection ............................................................................................. 22

4.1.1 Color Identification Algorithm ............................................................ 25

4.2 Local Search and Centroid Computation Algorithms ............................ 29

4.2.1 Initialization Phase ................................................................................. 29

4.2.2 Tracking Phase ....................................................................................... 34

CHAPTER 5  3D RECONSTRUCTION ........................................................................... 37

5.1 Triangulation Algorithm ................................................................................. 38

5.2 Point Correspondence: Straightforward Approach .............................. 40

5.3 Point Correspondence: Epipolar Geometry Approach .......................... 42

CHAPTER 6  HAND MODELING .............................................................................. 45

6.1 Skeleton Structure .......................................................................................... 47

6.2 Processing of Reconstructed Coordinates .................................................. 50

CHAPTER 7  PRACTICUM AND EXPERIMENTAL RESULTS ................................. 53

CHAPTER 8  CONCLUSIONS .................................................................................. 59

APPENDIX A  EPIPOLAR GEOMETRY ....................................................................... 61
LIST OF FIGURES

Figure 2.1a Virtual GloveboX at CGIPL, UNLV.................................
Figure 2.1b NASA's GloveboX prototype ...........................................
Figure 2.2 Modular software structure designed for UNLV's NASA VGX Project......................................................
Figure 3.1 Perspective Projection....................................................... 
Figure 3.2 The Camera Coordinate System with respect to the Global (World) Coordinate System.................................
Figure 3.3 Plane determined by the camera centers, the 3D space point and its projections onto the respective camera spaces........................
Figure 3.4 The image of the back-projection ray ($L_i$) determined by $\hat{p}_i$ in view 1 determines a line in the second view ($\ell_{12}$) containing the projection point $\hat{p}_2$ ......................
Figure 4.1 Projection of $\hat{p}$. All 3D points lying on the ray $L$ will be projected onto $\hat{p}$ ......................................................
Figure 4.2 Set-up of two cameras to provide stereo vision.................
Figure 4.3 Hand with color markers used for feature detection........
Figure 4.4 Spherical Coordinate Transformation..............................
Figure 4.5 Color triangle obtained after applying the SCT to the vectors in the RGB Color Cube.................................
Figure 4.6 Representation of an (r,g,b) vector using the Spherical Coordinates Transformation.................................
Figure 4.7 Ordering of the markers on the hand; the color around the indexes represents the color assigned to the respective marker........................................
Figure 4.8 Angles considered to properly sort the centroids closest to the black marker........................................
Figure 4.9 Finger with marker i at frame n........................................
Figure 5.1 Triangulation Method....................................................... 
Figure 5.2 The presence of noise in the image coordinates causes the projection rays to be skew ........................................
Figure 5.3 Estimation of the reconstructed 3D coordinates by minimizing the distance between the projection rays .......
Figure 5.4 The measured image points $\hat{p}_1$ and $\hat{p}_2$ are corrupted by noise; therefore they may not be a perfect match. The
points $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$ satisfy the epipolar constraint and minimize the cost function $\mathcal{C}(\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2)$.

Figure 6.1a Skeleton structure emulating the human hand
Figure 6.1b Skeleton as implemented in our 3D model
Figure 6.2a Polygonal mesh designed for the NASA VGX Project
Figure 6.2b Mesh with skin texture applied
Figure 6.3 Two different views (obtained with our software) of the local coordinate systems of each bone. Each local X axis is represented in red, the local Y axis in green, and the Z axis in yellow.

Figure 7.1a Examples of input images from Camera 1
Figure 7.1b Examples of input images from Camera 2
Figure 7.2 Results of the DFS pass (detected pixels marked in green) for the pair of input images in Figure 7.1
Figure 7.3 Centroids of the markers as computed with the data collected by the DFS sweep depicted in Figure 7.2
Figure 7.4 The shaded regions represent the reconstruction uncertainty region which grows larger as the angle between the projection rays becomes smaller.

Figure B.1a Original Image
Figure B.1b Color Histogram of the original image
Figure B.2a Processed image ($p = 85$)
Figure B.2b Expanded histograms of the modified image
Figure B.3a Processed image ($p = 75$)
Figure B.3b Expanded histograms of the modified image
Figure B.4 The blue dots represent the neighboring pixels that are inspected in order to determine whether the pixel of interest (black dot) is a strong candidate during the initialization phase
Figure B.5 Cases considered when inspecting a potential strong candidate (black dot) during the tracking phase
Figure C.1 Rotation about the X axis
ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Evangelos A. Yfantis, for his unfailing support and consistent guidance throughout my academic experience at UNLV.

I am very grateful to Dr. Ajoy Datta, Dr. Wolfgang Bein, and Dr. David Costa for taking time out of their busy schedules to be members of my committee.

I would like to thank my family, for they are an endless source of support, encouragement, and unconditional love. They mean everything to me, and none of this would have been possible without them by my side. To my brother Luis and my 'life-sister' Desiree I owe my being here writing these lines; their presence is the fuel of my life. Since there are not enough words for me to express my love and gratitude for my father, I should be satisfied with saying "I love you Dad, you are my world". I owe everything I am and everything I have accomplished to my wonderful family and my mother's ever-present fortitude. I also want to thank my cousins Luis and Gustavo Blando whose belief in me since day one made all of this possible. I will always be grateful for their support and unreserved help along all these years.
Last, but not least, I want to take the time to thank each and every one of my dear friends; they are always around to lend a hand or a pair of willing ears. To my dearest friend Ramzi Elkhater I want to thank for all his constant assistance and encouragement through thick and thin, as well as for his thorough proof-reading of this thesis. I would not have been able to make it out of this process in one piece without his unrelenting support and confidence in me. To Oksana Kubushyna for her readiness to help regardless of her busy and crazy schedule, thanks from the bottom of my heart, I would not have made it without your help, you are the coolest. I would like to thank my friends John Istle and Pamela Mandelbaum for their eagerness to help and listen. Special thanks go to the 'Young Kid from the West', also known as Kristal Sauer. Her contagious perkiness and good spirits brought new life to the project, and her willingness to learn made teaching her an enjoyable task. I will always be grateful to Kristal for her proficient, meticulous, and invaluable help in assembling and proof reading this manuscript.
CHAPTER 1

INTRODUCTION

Hand gestures are vital to human interaction, which endows the field of hand gesture recognition (HGR) with paramount importance in such applications as robot vision and man-machine interaction. Given the many fascinating applications and challenges present, a significant amount of research has been directed towards solving the problem of HGR. Some important considerations when planning an approach involve the selection of criteria for recognizing the hand. Also, it must be determined whether the algorithm should involve supplementary image preprocessing techniques.

The algorithm we will present in this thesis has been developed as part of the research in progress at the Computer Graphics and Image Processing Laboratory at the University of Nevada, Las Vegas. This research project is supported by NASA\(^1\), and its main goal is to develop an efficient application for Human-Computer Interaction (HCI) in virtual environments\(^2\) whose main premise is being non-invasive. Our virtual

\(^1\) NASA Space Grant/EPSCoR: “Development of a Nationally Competitive Program in Computer Vision Technologies for Effective Human-Computer Interaction in Virtual Environments”

\(^2\) For more information refer to http://biovis.arc.nasa.gov/vislab/vgx.htm
environment is modeled by using a custom-made box which is referred to as the Virtual GloveboX (VGX). Our aim is to develop a method in which the subject is unhampered by any kind of feedback-force device. In particular, we want to determine the motion of the hand solely from images taken from the movement of the hand where the subject wears, at most, a pair of ordinary gloves with color markers strategically located. In our approach, the 3D coordinates of the center of mass of the markers are determined and later used to render a 3D model of the hand. This is accomplished by means of the following tasks: video acquisition, marker detection, 3D reconstruction, and modeling. Our objective is to complete all these tasks with minimal error. Although more research is underway, the results so far obtained clearly show that our algorithm is quite robust despite the many potential sources of error.

The outline of this thesis is as follows. In Chapter 2, we will give an overview of our project and introduce the main components of our system. Chapter 3 includes background information necessary for the understanding of contents in later chapters. In Chapter 4, we delve into the specifics of the feature extraction phase, while Chapter 5 comprises the details of our 3D reconstruction algorithm. In Chapter 6, we briefly describe our 3D hand model as well as the connection between the feature extraction and the rendering of the hand model. Finally, Chapter 7 presents the reader with the practicum and experimental results; and in Chapter 8, we impart our conclusions.
CHAPTER 2

PROJECT DESCRIPTION

As mentioned above, the goal of the NASA Virtual Glovebox Project is to develop an efficient application for Human-Computer Interaction (HCI) in virtual environments whose main premise is being non-invasive. The VGX simulates the glove box that will be used in the International Space Station, which will soon provide a research facility for studying the long-term effects of microgravity on living systems. Since the VGX will be used to assist in earth-based training for these experiments, it has been designed to provide a realistic and fully immersive training environment.

The VGX system assembled by the UNLV CGIPL team consists of two components:

Hardware Component:

i. Glovebox,

ii. Cameras,

iii. Video Capture Card,

iv. Computer.

Software Component:

i. Video Capture Module,
ii. Feature Extraction Module,

iii. Hand Modeling Module.

The goal of our project is to determine the hand’s motion solely from two or more images showing its movement where the subject wears, at most, a pair of ordinary gloves with color markers positioned at the knuckles and the back of the hand. In order to extract the scene’s metric information from the pair of images, it is essential that the cameras be calibrated [1, 2, 3, 4, 5]. That is, all the intrinsic camera parameters, which determine the internal structure of the camera, as well as the extrinsic camera parameters, which are determined by the absolute position of the camera within our GloveboX must be computed. Thus, knowing the camera parameters allows us to reconstruct the 3D coordinates of the markers on the hand based exclusively on the knowledge of their locations in the images.

As part of our research, we developed a camera calibration algorithm tailored to our project’s needs. Our algorithm, which is based on the geometric properties of the camera and perspective projections, provides a closed form solution for the camera parameters. Thus, it requires low time and space complexity. Sufficient testing has also proven its high level of accuracy and robustness. An in-depth description of our calibration algorithm is given in [6].

In what follows, we will briefly describe the main characteristics of the Hardware and Software Components of our project.
2.1 Hardware Components

The UNLV CGIPL Virtual GloveboX was built according to NASA specifications and simulates the Virtual GloveboX used by NASA (Figure 2.1).

The project requires us to use multiple cameras. Thus, we will employ up to eight Pelco® CCC1370H-2 Series Digital CCD Cameras, each with a pixel array of 752x582.

(a) (b)

Figure 2.1: (a) Virtual GloveboX at CGIPL, UNLV. (b) NASA’s GloveboX prototype.

Since our ultimate goal is to reconstruct the 3D coordinates of the human hand, it is crucial that the cameras be synchronized since it is
assumed that both input images are taken at the same time and represent exactly the same 3D scene up to a certain degree of noise. Therefore, a four-real-time-video-digitizers 16-input video capture card provided by Statistical and Software Analysts Incorporated (SSAI)® was used to minimize the time delay between snapshots. The board provides real-time (30 frame per second (fps) for NTSC at a resolution of 640×480) frame-capturing capabilities for up to four simultaneous cameras. Due to an on-board synchronization circuit, the lag present in camera-to-camera switching is minimized. The board includes four independent video digitizers for high speed when working with multiple cameras. The design is based on the well-known Video Decoder Fusion878A from Conexant®, along with proprietary switching algorithms and image enhancement capabilities.

2.2 Software Components

Given the nature of the NASA VGX project, we have created an intricate system with a complex synchronization mechanism. In order to achieve component independency and real-time interaction, we designed several independent modules, each of which has a separate task hidden from the other modules. Since data exchange may eventually occur among modules, the system is designed so that inter-module communication is carried out via system-global dynamic data storage.

---

3 For more information refer to www.ssaigis
units implemented as queues. In fact, each module may have an input queue and an output queue (Figure 2.2).

![Diagram of modular software structure](image)

**Figure 2.2**: Modular software structure designed for UNLV's NASA VGX Project.

In this thesis, we will present a comprehensive description of the diverse components comprised in the Feature Extraction Module and a brief description of its interaction with the Hand Model Module.
CHAPTER 3

BACKGROUND INFORMATION

3.1 Camera Calibration

For the NASA VGX Project, we adopted the pinhole camera model [1, 7]. This is the most commonly used representation for devising a mathematical model for the camera and is based on the principles of perspective projections. Under this model, given a point \( \hat{\mathbf{x}} \) in space, its projection onto the image plane, say \( \mathbf{p} \), is obtained by intersecting the ray joining the point \( \hat{\mathbf{x}} \) and the center of projection \( \mathbf{C} \) (camera center) with the image plane as shown in Figure 3.1.

![Perspective Projection](image.png)

Figure 3.1: Perspective Projection.
Using triangle similarities, one can derive the mapping between $\hat{X}$ and $\hat{p}$. In fact, if $\hat{X}=(X,Y,Z)$, then $\hat{p}=(f\frac{X}{Z},f\frac{Y}{Z},f)$ on the image plane, where $f$ is the distance between the camera center $C$ and the principal point $O$ (which is the point of intersection between the principal axis and the image plane), and is referred to as the Focal Length. Thus, it follows that the mapping between world coordinates and image coordinates is given by:

$$
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} \rightarrow \begin{bmatrix}
f\frac{X}{Z} \\
f\frac{Y}{Z}
\end{bmatrix}
$$

(3.1)

To simplify the calculations, it is more convenient to express (3.1) using homogeneous coordinates (Appendix A.1). Indeed, the use of homogeneous coordinates (also known as projective coordinates) allows for coalescence of a large number of transformations as well as all symmetries of the plane. Consequently, all transformations considered can be regarded as linear maps in the space of triplets $[x_1,x_2,x_3]^T$ and can thus be expressed in terms of matrix multiplications. Accordingly, (3.1) can be rewritten as

$$
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} \rightarrow \begin{bmatrix}
fX \\
fY
\end{bmatrix} = \begin{bmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}.
$$

(3.2)
In order to yield a more precise model for the camera, let us note that in Figure 3.1, we assume not only that the axes of the image coordinates space have equal scales but also that the origin of coordinates in the image plane is at the principal point. However, this may not be the case. In fact, in most CCD models, the scale in the axial directions may not be the same, so two new parameters need to be introduced to account for this omission. Thus, letting \( s_x \) and \( s_y \) be the scale factors in the \( x \) and \( y \) directions, respectively, and \( (r_0,c_0) \) be the coordinates of the principal point, (3.2) becomes

\[
\begin{bmatrix}
X \\
Y \\
Z \\
1 
\end{bmatrix} \mapsto \begin{bmatrix}
f s_x (X+Zr_0) \\
f s_y (Y+Zc_0) \\
Z \\
1 
\end{bmatrix} = \begin{bmatrix}
f s_x & 0 & r_0 & 0 \\
0 & f s_y & c_0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 
\end{bmatrix} \begin{bmatrix}
X \\
Y \\
Z \\
1 
\end{bmatrix}. \tag{3.3}
\]

Hence, if we let

\[
K = \begin{bmatrix}
f s_x & 0 & r_0 & 0 \\
0 & f s_y & c_0 & 0 \\
0 & 0 & 1 & 0 
\end{bmatrix},
\]

we obtain

\[
\hat{p} = K \hat{X}, \tag{3.4}
\]

where \( K \) is referred to as the Camera Calibration Matrix and the parameters \( f, s_x, s_y, \) and \( (r_0,c_0) \) are referred to as the Intrinsic Camera Parameters.

Yet, our mathematical model is not accurate in view of the fact that, in Figure 3.1, we assume the points' coordinates to be expressed with
respect to the origin of the camera coordinate system, i.e., the camera center $C$. However, in general, the coordinates of the points in space will be given with respect to a global Euclidean coordinate frame. Nevertheless, this can be easily overcome since the global coordinate system and the camera coordinate system are related by means of a pair of linear transformations, namely, a rotation and a translation (Figure 3.2). Let $R$ and $t$ be the rotation and translation that take the global coordinate system into the camera coordinate system, respectively. These are referred to as the *Extrinsic Camera Parameters*.

![Figure 3.2: The Camera Coordinate System with respect to the Global (World) Coordinate System.](image)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In consequence, given \( \hat{X}_w \), a point in space with coordinates in the global coordinate system, its coordinates with respect to the camera coordinate system, denoted by \( \hat{X}_c \), are then given by
\[
\hat{X}_c = R \hat{X}_w + t.
\] (3.5)

Therefore, combining (3.4) and (3.5), it follows that
\[
\hat{p} = K \begin{bmatrix} R & t \end{bmatrix} \hat{X}_w,
\] (3.6)

where \( \hat{X}_w \) denotes the homogeneous coordinates of the three-dimensional point \( \hat{X}_w \). That is,
\[
\hat{p} = P \hat{X}_w,
\] (3.7)

where \( P = K \begin{bmatrix} R & t \end{bmatrix} \) is called the Projection Matrix.

In what follows,
\[
P^1 = K^1 \begin{bmatrix} R^1 & t^1 \end{bmatrix} \quad \text{and} \quad P^2 = K^2 \begin{bmatrix} R^2 & t^2 \end{bmatrix}
\] (3.8)

will denote the projection matrices for cameras 1 and 2, respectively.

A familiarity with the concept of epipolar geometry will be beneficial to the reader since it is of importance in our reconstruction algorithm. Thus, we describe it in the next section.

3.2 Epipolar Geometry

The epipolar geometry is the intrinsic projective geometry between two views [8, 9]. Among the most important characteristics of the epipolar geometry between two views are the facts that it depends only on the cameras' internal parameters and relative pose and that it is independent
of the scene structure. In what follows, a concise description of the most relevant concepts of the epipolar geometry will be presented; for in Chapter 5, it will be necessary to understand the relation between the projections of a point in space onto two different views.

Let $\hat{X}_w \in \mathbb{R}^3$, and $\hat{p}_1$ and $\hat{p}_2$ be its projections on Image 1 (corresponding to camera 1) and Image 2 (corresponding to camera 2), respectively. The goal is to find a relation between $\hat{p}_1$ and $\hat{p}_2$. Clearly, $\hat{X}_w$, $\hat{p}_1$, $\hat{p}_2$, and the camera centers $C_1$ and $C_2$ are coplanar (Figure 3.3); that is, they lie in the plane $\pi$ which is referred to as the epipolar plane.

![Figure 3.3: Plane determined by the camera centers, the 3D space point and its projections onto the respective camera spaces.](image-url)
Moreover, given that \( \hat{p}_i \) and \( \hat{p}_j \) are the projections of \( \hat{X}_w \), the rays emanating from the respective camera centers and passing through \( \hat{p}_i \) and \( \hat{p}_j \) not only intersect at \( \hat{X}_w \), but they also lie in \( \pi \).

We can also observe, from Figure 3.3, that the plane \( \pi \) can be uniquely determined by considering only one of the projection rays (provided one knows the coordinates of the 3D point and one of its projections) and the line segment connecting the camera centers, which is referred to as the baseline. Let \( L_i \) be the projection ray determined by \( \hat{p}_i \) and \( \hat{X}_w \). \( L_i \) is projected onto Image 2 as the line \( \ell_{12} \). Moreover, the image of \( \hat{X}_w \) in the view from camera 2, i.e. \( \hat{p}_j \), must lie on \( \ell_{12} \). Hence, if one assumes that \( \hat{p}_i \) is known, the search for its matching point on the second view is then limited to points lying on the epipolar line \( \ell_{12} \). Clearly, the same argument is valid for the reciprocal case, i.e., knowing \( \hat{p}_j \), its matching point in the first view can be found via a search along the epipolar line \( \ell_{21} \) (Figure 3.4).

---

4 Notation: \( \ell_{ij} \) denotes the epipolar line in the view from camera \( j \) determined by the projection ray through a given point in the view from camera \( i \).
Figure 3.4: The image of the back-projection ray ($L_1$) determined by $\hat{\mathbf{p}}_1$ in view 1 determines a line in the second view ($\ell_{12}$) containing the projection point $\hat{\mathbf{p}}_2$.

In Figure 3.4, the points $\epsilon_1 = (\epsilon_{11}, \epsilon_{12}, \epsilon_{13})$ and $\epsilon_2 = (\epsilon_{21}, \epsilon_{22}, \epsilon_{23})$ in view 1 and view 2, respectively, are the points of intersection of the baseline with the respective image planes. These points are referred to as the epipoles. It is important to note that all epipolar lines intersect at the epipoles. Moreover, any epipolar plane intersects both image planes in epipolar lines.

Therefore, by using the epipolar geometry of a pair of views, we reduce the search for matching points from a two-dimensional problem into a one-dimensional problem. This mapping from the two-dimensional projective space of view 1 into the pencil of epipolar lines through the
epipole $e_2$ in view 2 is represented by the so-called fundamental matrix, which will be denoted by $F$. The fundamental matrix can be derived algebraically in terms of the projection matrices of the stereo system, $P^1$ and $P^2$. This derivation is due to G. Xu and Z. Zhang [9] and will be concisely presented in what follows.

Let $\hat{p}_i$ be the projection point in view 1. We wish to determine the equation of its corresponding epipolar line $\ell_{12}$ in view 2. As mentioned above, $\ell_{12}$ is the projection onto image 2 of the back-projection ray $L_i$. However, $L_i$ is the ray containing all the points in space that project to $\hat{p}_i$; therefore, any point $\hat{X}$ lying on it satisfies (3.7), i.e., $\hat{p}_i = P^i \hat{X}$. In particular, $L_i$ can be thought of as the ray in space going through the points $C_i$ and $\hat{X} = (P^i)^\dagger \hat{p}_i$, where $(P^i)^\dagger$ is the right pseudo-inverse of the matrix $P^i$ (Appendix A.2). For simplicity, we will denote $(P^i)^\dagger$ by $(P^i)^\dagger$.

Therefore, $L_i$ is given by

$$\ell_i(\tau) = (P^i)^\dagger \hat{p}_i + \tau C_i, \, \tau \in \mathbb{R}^5. \quad (3.9)$$

Then, $\ell_{12}$ is determined by the projections via $P^2$ of the points $\hat{X} = (P^i)^\dagger \hat{p}_i$ and $C_i$. In fact,

$$\ell_{12} = (P^2 C_i) \times \left[ P^2 \left((P^i)^\dagger \hat{p}_i\right) \right]. \quad (3.10)$$

---

5 The equation of the line passing through points P and Q can be represented by the direction $P \times Q$. Thus, its equation is given by $(P \times Q)x, \, x \in \mathbb{R}^3$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Given that $P^2 \mathbf{c}_1 = \varepsilon_2$, (3.10) becomes

$$\ell_{12} = [\varepsilon_2]_x \left( P^2 \left( P^1 \right)^\top \hat{\mathbf{p}}_1 \right) = \left[ \varepsilon_2 \right]_x P^2 \left( P^1 \right)^\top \hat{\mathbf{p}}_1,$$

(3.11)

where $[\varepsilon_2]_x$ is defined by

$$[\varepsilon_2]_x = \begin{bmatrix} 0 & -\varepsilon_{23} & \varepsilon_{22} \\ \varepsilon_{23} & 0 & -\varepsilon_{21} \\ -\varepsilon_{22} & \varepsilon_{21} & 0 \end{bmatrix},$$

(Appendix A.3). That is,

$$\ell_{12} = F \hat{\mathbf{p}}_1,$$

(3.12)

where

$$F = \left[ [\varepsilon_2]_x P^2 \left( P^1 \right)^\top \right]$$

(3.13)

is the fundamental matrix. Clearly, $F$ is the matrix representing the mapping $F : \hat{\mathbf{p}}_1 \mapsto \ell_{12}$.

From (3.8), we know that $P^1 = K^1 \left[ \mathbf{R}^1 \mid \mathbf{t}^1 \right]$ and $P^2 = K^2 \left[ \mathbf{R}^2 \mid \mathbf{t}^2 \right]$. However, to simplify the calculations, one can assume, without loss of generality, that the origin of the global coordinate system is at $C_1$, i.e., the optical center of camera 1. Then, it follows that

$$P^1 = K^1 \left[ \mathbf{I}_3 \mid \mathbf{0}_{3 \times 1} \right], \quad P^2 = K^2 \left[ \mathbf{R} \mid \mathbf{t} \right] \quad \text{and} \quad C_1 = \begin{pmatrix} \mathbf{0}_{3 \times 3} \\ 1 \end{pmatrix},$$

(3.14)

where $\mathbf{R} = \mathbf{R}^1 \left( \mathbf{R}^2 \right)^{-1}$, $\mathbf{t} = \mathbf{t}^1 - \mathbf{R}^1 \left( \mathbf{R}^2 \right)^{-1} \mathbf{t}^2$, $\mathbf{I}_3$ is the identity matrix of dimensions $3 \times 3$, and $\mathbf{0}_{3 \times 1}$ is the null column vector. Then, according to (3.13), we have that

17
Next, assume the points \( \hat{p}_1 \) and \( \hat{p}_2 \) are matching image points. Then, we know that \( F \hat{p}_1 = \ell_{12} \) is the epipolar line corresponding to \( \hat{p}_1 \) in view 2 and that \( \hat{p}_2 \in \ell_{12} \). Hence, \( \hat{p}_2^T \ell_{12} = 0 \); that is

\[
\hat{p}_2^T F \hat{p}_1 = 0.
\]

Moreover, if \( \hat{p}_1 \) and \( \hat{p}_2 \) are image points satisfying (3.16), the rays determined by these points are coplanar, which is a necessary condition for points to correspond. Equation (3.16) represents one of the most important properties of the fundamental matrix and is referred to as the epipolar constraint.

In the context of the NASA VGX, we initialize our system by calibrating the cameras. Therefore, we know a priori all the camera’s parameters – in particular, the camera calibration matrix \( K \). From (3.6), we know that \( \hat{p} = K \begin{bmatrix} R & t \end{bmatrix} \hat{x}_w \); thus, if \( K \) is known, we have that \( K^{-1} \hat{p} = \begin{bmatrix} R & t \end{bmatrix} \hat{x}_w \) (note that \( K \) is invertible). The point \( \bar{p} = K^{-1} \hat{p} \) is the image point referred to as the normalized coordinates of \( \hat{p} \). Similarly, the camera matrix \( K^{-1} P = \begin{bmatrix} R & t \end{bmatrix} \) is called the normalized camera matrix. Therefore, if we carry out the derivation of the fundamental matrix with the normalized camera matrices \( P^1 = \begin{bmatrix} I_3 & 0_{3,1} \end{bmatrix} \) and \( P^2 = \begin{bmatrix} R & t \end{bmatrix} \), we obtain the matrix

\[
F = \left(K^2\right)^{-T} \begin{bmatrix} t \end{bmatrix} \begin{bmatrix} R(K^1)^{-1} \end{bmatrix}^{-1}
\]

(3.15)
\[ E = [t]_R, \]  

which is called the essential matrix. As with the fundamental matrix, the defining equation for the essential matrix is given by

\[ \vec{p}_1^T E \vec{p}_1 = 0, \]  

for any pair of corresponding normalized image points \( \vec{p}_1 \) and \( \vec{p}_2 \).

Note that since \( \vec{p}_1 = (K^1)^{-1} \hat{p}_1 \) and \( \vec{p}_2 = (K^2)^{-1} \hat{p}_2 \), (3.18) yields

\[
0 = \left[ (K^2)^{-1} \hat{p}_2 \right]^T E (K^1)^{-1} \hat{p}_1 \\
= \vec{p}_2^T (K^2)^{-T} E (K^1)^{-1} \hat{p}_1
\]

for any pair of corresponding image points \( \hat{p}_1 \) and \( \hat{p}_2 \). Then,

\[
F = \left( K^2 \right)^T E \left( K^1 \right)^{-1} \iff E = \left( K^2 \right)^T F K^1.
\]  

This completes our overview of epipolar geometry, and we shall now ensue to a description of the image acquisition stage.
FEATURE EXTRACTION

CCD cameras acquire images by effectively performing a perspective projection of the scene under observation (i.e., the three-dimensional world) onto the image plane, as explained in the previous chapter. As a consequence, if the camera has been calibrated, we are able to compute the inverse projection which will ultimately allows us to translate image coordinates into 3D world coordinates.

In perspective projection, however, infinitely many different 3D points will be projected onto the same point in the image. As Figure 4.1 shows, the point $\hat{X}_i$ is projected through the pinhole to $\hat{p}$ in the image, but any point along the back-projection ray $L$ is also projected to $\hat{p}$. Thus, it is not possible to reconstruct the 3D coordinates of the point projected to $\hat{p}$ by using only one calibrated camera, for we cannot infer its depth. Nonetheless, this problem can be overcome by using at least two calibrated cameras, as shown in Figure 4.2.
Figure 4.1: Projection of $\hat{p}$. All 3D points lying on the ray $L$ will be projected onto $\hat{p}$.

Figure 4.2: Set-up of two cameras to provide stereo vision.

Due to this fact and the nature of our project, our system will consist of multiple cameras (at least eight) so that we can have stereo vision at all times regardless of the position of the user's hand within the box.
However, in the first stage of our research, our set-up consists of only two cameras (Figure 4.2). As mentioned in Chapter 2, each pair of images (one image per camera) must represent, up to a certain degree of noise, the scene at the same point in time. This synchronization between frames is performed by the video capture card (Section 2.1).

With this setting we must then proceed with the first of the tasks towards the 3D reconstruction, that is, we must detect and track the markers on the user's glove. However, before the marker tracking can be carried out we must initialize the tracking system. All these tasks are included in the Marker Detection algorithm to be presented next.

4.1 Marker Detection

As we will show in Chapter 6, in order to mimic the movement of the user's hand with our 3D Hand Model, we need to obtain the directional cosines of each phalanx. To determine these coordinates, we chose to use an array of color markers placed on the back of the palm and the joints of the fingers. In order to comply with the requirement of our approach being non-invasive, we use a pair of regular rubber gloves. Thus, the markers are not directly attached to the hand, and no external feedback force device needs to be used (Figure 4.3).
The complexity of the overall project requires that, at any given point in time, several different tasks be performed. Since we want our solution to operate real-time, we must at all times minimize the computational load. One step towards accomplishing this goal was to design our system so that difficulties inherent to the color detection problem, such as background interference, shadows, changes in lighting, etc., were diminished. Hence, we chose a matte black background which provides a simple backdrop and eliminates shadows of the hand and any other surrounding objects.

The next step was to choose the colors to be used on the glove so that the tracking stage would be not only successful but also efficient. Theoretically, the color values of red, green, yellow, and blue are totally distinct. However, in practice, many sources of error, especially changes
in lighting and capture settings, introduce noise that precludes acquisition of precise values. Furthermore, given that the user's hand is in constant motion within the GloveboX, the markers appear to be different shades of their original color based on their position relative to the light and the cameras. Thus, after experimenting with different colors exposed to diverse settings, we established that the most resilient colors were red and green. Because of this and the fact that our ultimate goal is to estimate any possible gesture of the hand, we assigned each finger either red or green in an alternating fashion. Given that, in order to conform to the data requirements of our three-dimensional hand model, we needed to place a marker on the back of the hand, which is adjacent to all the knuckles and can, therefore, be neither red nor green, we chose to use a black marker. The remaining two markers on the back of the hand are near the wrist and far from the knuckles which allowed us to color them red and green. It is important to note that the color of the glove (white) surrounds the markers, thereby preventing the use of black for the back of the hand from conflicting with the background.

Once the colors have been assigned to each finger, the markers are placed so that their centers lie directly over the center of the specified joint. As a result, after detecting all the marker's constituent pixels, we can compute the coordinates of its center of mass, i.e., its centroid, which are then used as the image coordinates of the corresponding joint.
Another step towards maintaining the manageability of the computational load is that of minimizing the amount of image preprocessing. We accomplish this in large part by the design of our system. Yet, in order to improve the color detection even further, we decided to include an equalization stage (Appendix B.1) prior to the detection stage [10, 11, 12]. This phase significantly enhances the color detection without compromising the time complexity of the overall computation.

Finally, we reduce the searching area for the detection stage by initializing the tracking system. This is done by computing the centroids of the markers at a predetermined position within the VGX (Section 4.2.1). Thus, the area to be examined while searching for characteristic pixels is not the entire image but a neighborhood of the reference position. Once this is accomplished, the marker detection phase is carried out. The latter works based on a number of different tasks, namely, color identification, local search (for each marker/centroid), and a Depth-First Search (DFS) sweep to accurately compute the coordinates of each of the new centroids for the current frame [17, 18]. In what follows, we will describe each of these tasks in detail.

4.1.1 Color Identification Algorithm

Color images are modeled so that the image data is divided into three monochrome bands. Hence, the information stored in a digital image is the brightness information in each spectral band. Normally, images are
represented as red, green, and blue (RGB) images which have 24 bits per pixel (8-bit monochrome standard) [13].

Given the complexity of the task of detecting color in images, it is necessary in many applications to process the RGB information so that the brightness and color information are decoupled from it. By doing so, we obtain two new spaces: a one-dimensional space containing the brightness information and a two-dimensional space containing the color information. There exist several decoupling transformations, many of which are geometric approximations to mapping the RGB color cube into some other color space. The choice of transformation depends on the application itself. Given that one of the priorities for the VGX project is that of minimum computational load, we decided to implement a color transformation solely based on the geometry of the RGB space. In fact, we performed a Spherical Coordinate Transform (SCT) on the RGB space. This transform is purely geometrical and computationally inexpensive.

Then, given an RGB vector \((r, g, b)\), its transformed coordinates are given by

\[
\begin{align*}
L &= \sqrt{r^2 + g^2 + b^2} \\
\theta &= \cos^{-1} \left( \frac{b}{L} \right) \\
\psi &= \tan^{-1} \left( \frac{g}{r} \right)
\end{align*}
\]

(4.1)
where $L$ is the modulus of the RGB vector, $\theta$ its angle of deflection from the blue axis, and $\psi$ the angle of deflection of its projection on the RG-plane from the red axis, as shown in Figure 4.4.

![Figure 4.4: Spherical Coordinate Transformation](image)

By applying the SCT, the RGB space is transformed into a two-dimensional color space defined by the angles $\theta$ and $\psi$, and a one-dimensional brightness space defined by $L$. This space can be represented by a color triangle (Figure 4.5) so that color can be defined as the point in the triangle that the vector passes through, as shown in Figure 4.6.
Figure 4.5: Color triangle obtained after applying the SCT to the vectors in the RGB Color Cube

Figure 4.6: Representation of an (r,g,b) vector using the Spherical Coordinates Transformation.
Since we are interested in the color characteristics of the pixels, we disregard the brightness information. Nonetheless, given the different shades of color that may be present, the information obtained from the $\theta$ and $\psi$ components is not enough to discern the color of a given pixel. To surmount this difficulty, we also consider the saturation value of the RGB vector, which will be denoted by $s$. That is,

$$s = 1 - 3 \left( \frac{\min(r, g, b)}{r + g + b} \right).$$

(4.2)

Saturation is a dimension of color affecting our response to it since it refers to a color's purity. Indeed, it is the degree to which a color is undiluted by white light. Hence, a highly saturated color contains no white light, making it appear deeper and purer than a similar but less saturated color. On the other hand, if a color has no saturation, it appears as a shade of gray.

Thus, the mapping $(r, g, b) \mapsto (\theta, \psi, s)$ allows us to efficiently identify all different shades of color present in the markers at any point in time. Moreover, the aforementioned transformation has proven to be robust and resilient to changes in the setting of the system.

4.1.2 Local Search and Centroid Computation Algorithms

4.1.2.1 Initialization Phase

This phase provides the image coordinates of the centroid of each marker while the user positions his hand at a known location within the GloveboX. The output of this phase will be referred to as the reference...
frame. Before we proceed, we wish to note that the markers on the glove are identified by the algorithm in the order shown in Figure 4.7.

Figure 4.7: Ordering of the markers on the hand; the color around the indexes represents the color assigned to the respective marker.

To enhance readability, we will denote the centroids of the markers by \( m_{i,n} \), where \( i \) represents the index of the marker on the glove and \( n \) the number of the frame being processed. Thus, when referring to the reference frame, \( n \) should be zero, and the set of centroids will be
We may now present the steps involved in the initialization of the tracking system.

In order to reduce the amount of processing, a search region is determined (a rectangle) such that it enfolds the entirety of the glove. This region is chosen sufficiently large so as to provide the user with certain flexibility when positioning the hand within the GloveboX.

The search for the centroids begins once the user has selected an arbitrary point within the black marker in both images. This is necessary given that the background is black; otherwise, the search would mistakenly single out points on the background as constituents of the black marker. Then the selected point is used to efficiently compute the coordinates of the black centroid by means of a DFS starting at the specified location. The coordinates of this centroid will also serve as a sieve to determine whether a pixel belongs to a marker or not, by virtue of the fact that all markers on the hand satisfy particular distance and positioning requirements with respect to $m_{0,0}$.

In order to locate the red and green markers, we examine the search region in a row by row manner seeking for what we refer to as a *strong candidate*, that is, a pixel that has the potential to be part of one of the color markers. For the purpose of the initialization phase, a strong candidate is a pixel, detected to be either red or green, which is completely surrounded by pixels of its same color (Appendix B.2) and lies
within a specified distance from \( m_{0,0} \). We wish to point out that the color of each pixel under observation is determined by means of the aforementioned color identification algorithm (Section 4.1). Once a strong candidate is located, a DFS implementing the abovementioned color identification algorithm is performed, thus obtaining all the constituent pixels of that particular marker, which then allows us to compute the coordinates of its centroid.

The inspection of the search region results in the successful location of all the centroids of the markers on the glove. The centroids are grouped by color (red or green); however, they are not in the order depicted in Figure 4.7. In order to sort them properly, we first locate those centroids that are closest to the black centroid, namely, \( m_{1,0} \), \( m_{5,0} \), \( m_{9,0} \), \( m_{13,0} \), \( m_{17,0} \), and \( m_{18,0} \) (Figure 4.7). By doing so, sorting the remaining centroids becomes considerably easier given that all centroids of markers lying on the same finger present a small discrepancy in their \( x \) coordinates.

To facilitate the sorting of the centroids just named, we compare the angles determined by the vertical axis passing through \( m_{0,0} \) and the line segment joining the centroid being inspected and \( m_{0,0} \), as shown in Figure 4.8.
Figure 4.8: Angles considered to properly sort the centroids closest to the black marker.

This categorization is possible in view of the fact that the position of the centroids previously referenced remains fixed despite the movement of the hand; therefore, the angles remain constant. This last step yields the sought-after ordering of the centroids (Figure 4.7).

In addition to providing the location of all the markers at the beginning of the video sequence, the initialization stage allows us to
compute the directions of all the phalanxes, which will be vital for the realization of the tracking phase.

4.1.2.2 Tracking Phase

As mentioned earlier, prior to the marker detection algorithm, the system is initialized as described in Section 4.1.2.1. Thus, in subsequent frames and as the user moves his hand, the location of each marker is tracked based on the knowledge of its position in the previous frame. To accomplish this, specific information regarding the position of the marker is stored at each frame. In what follows, we give a thorough explanation of the local search.

For simplicity, and without loss of generality, let us suppose we are processing the images obtained for the \( n \)th frame and that we are interested in locating the new position of marker \( i \), which we denote \( m_{i,n} \), (i.e., we have already tracked all markers with indices 0 to \( i-1 \)), which lies at one of the finger's two internal joints (Figure 4.9). Our first task is to find the best candidate \( \hat{p}_i \) at which to launch the local search. The image coordinates of \( \hat{p}_i \) are estimated utilizing the information stored about the centroids of markers \( m_{i,n-1} \) and \( m_{i-1,n-1} \) in the previous frame. If \( m_{i,n} = (x_{i,n}, y_{i,n}) \), then we proceed with the search for the candidate by letting

\[
\hat{p}_{i,n} = m_{i-1,n} + [m_{i,n-1} - m_{i-1,n-1}].
\]  

(4.3)
Now that we have an estimate of marker $i$'s new location, we must determine whether $\hat{p}_i$ is a strong candidate. We accomplish this by inspecting $\hat{p}_i$'s neighboring pixels in search of pixels of the color assigned to $m_{i,n}$. For tracking purposes, a pixel is a strong candidate if at least four pixels of the required characteristics are found (Appendix B.2). Otherwise, we use $\hat{p}_i$ to find a better candidate by scrutinizing the pixels in a neighborhood of $\hat{p}_i$ of radius one and systematically incrementing the radius of search until a strong candidate is found. Then, if $\hat{p}_i$ is designated as the strong candidate, we proceed to inspect the area around it by means of the DFS routine that implements the color identification algorithm (Section 4.1.1) to identify all the
constituents pixels of marker $i$. These are in turn employed in the 
computation of the image coordinates of the marker's centroid.

It is important to note that, in the case of $\hat{p}_i$ being a *weak candidate*, 
the systematic search for a stronger candidate in gradually larger 
neighborhoods of $\hat{p}_i$ never exceeds a radius of search larger than five. 
This can be attributed to the fact that at 30 frames per second (fps) the 
discrepancy between frames is trifling. Thus, a strong candidate will not 
lie further than five pixels away from $\hat{p}_i$ (this has proven to be true even 
at 15 fps).

The procedure just depicted is performed, at frame $n$, on the images 
from camera 1 and camera 2 and the respective image coordinates of the 
point $m_{in}$ that it yields will be denoted by $\tilde{p}_{i1}^n$ and $\tilde{p}_{i2}^n$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
CHAPTER 5

3D RECONSTRUCTION

3D reconstruction is the most important step towards reconstructing the pose of the human hand since it provides the three-dimensional information required to estimate the object’s position and pose. As mentioned above, before the reconstruction phase can be carried out, it is essential that the cameras be calibrated. As a result of the camera calibration process, the positions of each camera with respect to a global coordinate system (rotation and translation matrices), as well as their internal structures, are obtained [1, 3, 4, 6].

As stated in Chapter 4, at least two views are necessary to resolve for the three coordinates of a point from its projection onto both images. However, in order to reconstruct the 3D coordinates of the observed point, the first step is to match the points in the two images. Indeed, both image points must represent the same point on the scene at the same instant in time. Once the points are matched, 3D reconstruction is performed by means of a triangulation algorithm. In this chapter, we will present two approaches to the reconstruction problem in the context of the NASA VGX project. The experimental results obtained with each
approach will be presented in Chapter 7. Let us begin by depicting the triangulation algorithm.

5.1 Triangulation Algorithm

The goal of triangulation is to obtain the point of intersection of the back-projection rays corresponding to a pair of matching image points, say $\mathbf{p}_1$ and $\mathbf{p}_2$, as shown in Figure 5.1.

![Figure 5.1: Triangulation method.](image)

In order to simplify the calculations, we can assume, without loss of generality, that the origin of the global coordinate system is located at the camera center $\mathbf{C}_1$. Then, according to (3.14), we have that

\[
P^1 = K^1 \left[ I_3 \mid \mathbf{0}_{3 \times 1} \right], \quad P^2 = K^2 \left[ \mathbf{R} \mid \mathbf{t} \right], \quad \text{and} \quad \mathbf{C}_1 = \begin{pmatrix} \mathbf{0}_{3 \times 3} \\ 1 \end{pmatrix}. \quad (5.1)
\]

38

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Thus, if \( r_1(\tau) \) and \( r_2(\gamma) \) are the back-projection rays corresponding to \( \hat{p}_1 \) and \( \hat{p}_2 \), respectively, their equations are given by

\[
r_1(\tau) = \tau \left( P^1 \right)^\dagger \hat{p}_1, \quad \tau \in \mathbb{R}, \tag{5.2}
\]

and

\[
r_2(\gamma) = \gamma \left( P^2 \right)^\dagger \hat{p}_2, \quad \gamma \in \mathbb{R}. \tag{5.3}
\]

Then, the line in space that intersects both rays perpendicularly has equation

\[
r_3(\delta) = \delta \left[ \left( \left( P^1 \right)^\dagger \hat{p}_1 \right) \times \left( \left( P^2 \right)^\dagger \hat{p}_2 \right) \right] + \tau \left( P^1 \right)^\dagger \hat{p}_1, \tag{5.4}
\]

for some \( \tau \in \mathbb{R} \) and \( \delta \in \mathbb{R} \); and the point of intersection of all three lines is the solution to the linear system

\[
\delta \left[ \left( \left( P^1 \right)^\dagger \hat{p}_1 \right) \times \left( \left( P^2 \right)^\dagger \hat{p}_2 \right) \right] + \tau \left( P^1 \right)^\dagger \hat{p}_1 - \gamma \left( P^2 \right)^\dagger \hat{p}_2 = 0. \tag{5.5}
\]

Solving (5.5), we obtain the three-dimensional coordinates of the unique point \( \hat{X} \) satisfying \( \hat{p}_1 = P^1 \hat{X} \) and \( \hat{p}_2 = P^2 \hat{X} \).

In the context of the NASA VGX, completing the triangulation stage is equivalent to having the reconstructed 3D coordinates of the centroids of the markers on the glove. As we will show in Chapter 6, this three-dimensional data is fed into the Hand Modeling Module which utilizes it to model the 3D hand accordingly.
In what follows, we describe two ways in which, for each of the markers on the glove, we selected the pair of matching image points $\hat{p}_1$ and $\hat{p}_2$ that would be used for the triangulation procedure.

5.2 Point Correspondence: Straightforward approach

To facilitate the 3D reconstruction, it is necessary to have a set of image point correspondences. For this particular approach, the map of point correspondences is the one obtained from the Marker Detection Algorithm (Chapter 4). It is worth noting that since the detection of the markers was performed in order (Figure 4.6), the matching of the centroids is directly accomplished via the matching of their indices. Subsequently, those correspondences are supplied to the triangulation algorithm described in Section 5.1 which yields the sought-after 3D coordinates of the markers on the glove.

Still, in view of the fact that the pairs of image coordinates obtained from the marker detection algorithm, $\left\{\left(\hat{p}_{u_0}, \hat{p}_{v_0}\right)\right\}_{0 \leq i \leq 18}$, are corrupted with noise, the corresponding back-projection rays will be skew. As a consequence, given a pair $\left(\hat{p}_{u_0}, \hat{p}_{v_0}\right)$, $0 \leq i \leq 18$, there will be no three-dimensional point $\hat{X}_i$ whose projected image coordinates coincide with those in the given pair (Figure 5.2).
Figure 5.2: The presence of noise in the image coordinates causes the projection rays to be skew.

Nevertheless, we can estimate the 3D coordinates of the reconstructed point associated with the pair \( (\hat{p}_u, \hat{p}_v) \), \( 0 \leq i \leq 18 \), by considering \( \hat{X} \) to be the point in space that minimizes the Euclidean distance between the two back-projection rays, as shown in Figure 5.3.

Figure 5.3: Estimation of the reconstructed 3D coordinates by minimizing the distance between the projection rays.
5.3 Point Correspondence: Epipolar Geometry Approach

In the approach to the Point Correspondence problem illustrated in the previous section, we assumed the back-projection rays will be skew (due to the noise in the data), and we estimated the coordinates of the 3D point by minimizing the distance between the back-projection rays obtained from the “noisy pair”. However, instead of considering image points that may not be a perfect match, we can use the “noisy pair” \((\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2)\) along with the epipolar constraint in (3.16) to find a new pair of corresponding points \((\hat{\mathbf{p}}'_1, \hat{\mathbf{p}}'_2)\) such that \(\hat{\mathbf{p}}'_2 F \hat{\mathbf{p}}'_1 = 0\), i.e., \(\hat{\mathbf{p}}'_1\) and \(\hat{\mathbf{p}}'_2\) are a perfect match. Then, the reconstructed point can be found by the triangulation process expounded upon in Section 5.1, for the back-projection rays will intersect precisely in space.

Since the new image points \(\hat{\mathbf{p}}'_1\) and \(\hat{\mathbf{p}}'_2\) should lie in a neighborhood of the measured points \(\hat{\mathbf{p}}_1\) and \(\hat{\mathbf{p}}_2\), our goal is to solve the minimization problem

\[
\min_{\hat{\mathbf{p}}'_1, \hat{\mathbf{p}}'_2} \mathcal{L}(\hat{\mathbf{p}}'_1, \hat{\mathbf{p}}'_2) \quad \text{subject to} \quad \hat{\mathbf{p}}'_2 F \hat{\mathbf{p}}'_1 = 0, \tag{5.6}
\]

where \(\mathcal{L}(\hat{\mathbf{p}}'_1, \hat{\mathbf{p}}'_2) = \|\hat{\mathbf{p}}'_1 - \hat{\mathbf{p}}_1\|^2 + \|\hat{\mathbf{p}}'_2 - \hat{\mathbf{p}}_2\|^2\) (Figure 5.4).

In (5.6), \(\hat{\mathbf{p}}'_2 F \hat{\mathbf{p}}'_1 = 0\) so that both image points must lie on a pair of corresponding epipolar lines, namely, \(\ell_{12}\) and \(\ell_{21}\) (Figure 3.4). On the other hand, we also know that any pair of image points lying on
corresponding epipolar lines will satisfy the epipolar constraint. In particular, the pair of corresponding points \( \hat{p}_1 \) and \( \hat{p}_2 \) which lie on \( \ell_{12} \) and \( \ell_{21} \), respectively, minimizes the distance to the measured points. But then, \( \hat{p}_1 \) and \( \hat{p}_2 \) are the points that minimize the distance between the measured points and the respective epipolar lines.

Figure 5.4: The measured image points \( \hat{p}_1 \) and \( \hat{p}_2 \) are corrupted by noise; therefore they may not be a perfect match. The points \( \hat{p}_1' \) and \( \hat{p}_2' \) satisfy the epipolar constraint and minimize the cost function \( \mathcal{C}(\hat{p}_1, \hat{p}_2) \).
Hence, given that \( \| \mathbf{p}_1 - \mathbf{p}_1' \|_2 = d^2(\mathbf{p}_1, \ell_{21}) \) and \( \| \mathbf{p}_2 - \mathbf{p}_2' \|_2 = d^2(\mathbf{p}_2, \ell_{12}) \), where \( d(Q, \ell) \) represents the Euclidean distance from a point to a line, the minimization problem in (5.6) can be restated as

\[
\min_{\ell_{21}, \ell_{12}} C(\mathbf{p}_1, \mathbf{p}_2), \tag{5.7}
\]

where \( C(\mathbf{p}_1, \mathbf{p}_2) = d^2(\mathbf{p}_1, \ell_{21}) + d^2(\mathbf{p}_2, \ell_{12}) \), and the minimum is considered over all possible epipolar lines. Now that the minimization is carried out over a set of lines, one can further reduce the computations. The pencil of epipolar lines is parameterized so that they can all be represented in terms of a single parameter, say \( \alpha \in \mathbb{R} \). Then, if \( \ell_{21} : \ell_{21}(\alpha) \) and \( \ell_{12} : \ell_{12}(\alpha) \), (5.7) becomes

\[
\min_{\alpha \in \mathbb{R}} \left[ d^2(\mathbf{p}_1, \ell_{21}(\alpha)) + d^2(\mathbf{p}_2, \ell_{12}(\alpha)) \right]. \tag{5.8}
\]

Thus, the minimization problem in (5.6) has been reduced to a one-dimensional minimization problem which can be solved using basic calculus and algebra techniques [7].
CHAPTER 6

HAND MODELING

The three-dimensional hand model developed for the NASA VGX project consists of two parts, a skeleton and a mesh [14, 15]. The skeleton is a set of connected bones, where each bone can be rotated individually according to its assigned degrees of freedom. This set of bones has a hierarchical structure, where each bone has a parent bone and a child bone (Figure 6.1). From this definition and Figure 6.1, it follows that we have five skeletons \([S_0,...,S_5]\) (one for each finger), each of which is constituted by bones \(B_0, B_1, \) and \(B_2\). That is, for \(0 \leq i \leq 4\), \(S_i = \{B_{i0}, B_{i1}, B_{i2}\}\).

The purpose of the mesh is to endow the model with a more lifelike appearance. The mesh consists of a set of polygons which are rendered as solid lit surfaces with applied texture and color that simulate real human skin [16]. Together, geometry, texture, and lighting create a realistic effect, as can be seen in Figure 6.2. Once the skeleton and the mesh are bound together, the result is an interactive 3D model of a human hand.
In order to understand how the reconstructed coordinates of the markers are utilized to emulate the movement of the hand with the 3D model, one must understand the underlying structure of the model. In what follows, we will present a concise description of the skeleton structure. However, those interested in delving deeper into all the facets of our 3D hand model are encouraged to refer to [14].
6.1 Skeleton Structure

Each bone is created within its own local coordinate system. In addition to its direction, each bone must "know" its parent and its child, i.e., the bones to which it is connected. The bone joined at the tip is considered its "child", and the bone joined at the base is considered its "parent". The child is always dependent on its parent in the sense that the child's local coordinate system is constructed according to the parent's direction (Figure 6.3).

Figure 6.2: (a) Polygonal mesh designed for the NASA VGX Project. (b) Mesh with skin texture applied.
Figure 6.3: Two different views (obtained with our software) of the local coordinate systems of each bone. Each local X axis is represented in red, the local Y axis in green, and the Z axis in yellow.

For example, consider a bone chain $B_0...B_n$, where $B_0$ is the root (first parent) and $B_n$ is the tip (last child). Then, each child $B_k$ is defined recursively as

$$\tilde{T}_k = \tilde{T}_{k-1} T_k R_k,$$

(6.1)

where $T_k$ is the translation matrix associated to bone $B_k$, which sets the center of rotation to the base of the bone; $R_k$ is the rotation matrix associated to bone $B_k$; and $\tilde{T}_{k-1}$ is the accumulated transformation matrix from the operations on the previous bones, with $\tilde{T}_0 = I$.
Also, the directional cosines of each bone are used to represent its direction. That is, if $\theta_x$, $\theta_y$, and $\theta_z$ are initial rotation angles of the bone, then

\[
\begin{align*}
L &= \sqrt{\cos^2\theta_x + \cos^2\theta_y + \cos^2\theta_z} \\
\cos_X &= \frac{\cos\theta_x}{L} \\
\cos_Y &= \frac{\cos\theta_y}{L} \\
\cos_Z &= \frac{\cos\theta_z}{L}
\end{align*}
\] (6.2)

The next step is to understand how each bone is moved. For a bone to be rotated, two pieces of information are required. Indeed, it is required that we know the axis of rotation about which to rotate the bone, along with the angle of rotation by which the bone should be rotated. Given that any particular rotation can be seen as a sequence of rotations about the X, Y, and Z axes, the procedure can be regarded as a series of rotations about local axes (Appendix C). Let $\phi^k_x$, $\phi^k_y$, and $\phi^k_z$ be the current rotations describing the movement of bone $B_k$ in the chain $B_0...B_n$. If $R(\phi^k_x, \phi^k_y, \phi^k_z)$ is the current rotation matrix associated with $B_k$ (Appendix C), then the composite transformation matrix at bone $B_k$ is given by

\[
\hat{T}_k = \hat{T}_{k-1} \ T_k \ R(\phi^k_x, \phi^k_y, \phi^k_z).
\] (6.3)

Before we continue, it is important to note that in order for the skeletal structure to realistically represent the movements of a human...
hand, certain constraints need to be added. In fact, (6.3) can model a movement for any triplet of angles \((\phi_x^k, \phi_y^k, \phi_z^k)\). As a consequence, the skeleton is capable of representing infinitely many different states of the hand, many of them movements that would be impossible for a human to perform. Therefore, a set of constraints is designed so as to limit the range of motion of each bone; thus eliminating unrealistic bone configurations [14]. As one last remark regarding the set-up of the hand model, we wish to mention that, although each bone has its own local coordinate system, the global coordinate system (through which the data must be processed) has its origin at the base of bone \(B_{10}\) (Figure 6.1). Moreover, the global coordinate system, set by the canonical orthonormal base in \(\mathbb{R}^3\), determines the relative position of the palm of the hand in space.

6.2 Processing of Reconstructed Coordinates

From the 3D Reconstruction stage, we obtain a set of 3D coordinates corresponding to each of the markers on the user's glove. According to the notation in Chapter 5, let \(\mathbf{X}_i\) be the reconstructed 3D coordinates of marker \(i\) in the \(n^{th}\) frame. The first step is to translate the coordinates set \(\{\mathbf{X}_i\}_{i=0,\ldots,18}\) to the origin of the global coordinate system. Subsequently, we must determine the new coordinate system for the palm of the hand so that we can then obtain the linear transformation necessary to change bases from the actual coordinate frame to the global
coordinate frame. To that effect, the new X axis is determined by the normalized direction vector \( \left( \hat{X}_{17,n} - \hat{X}_{18,n} \right) \), and the new Y axis is determined by the direction vector \( \left( \hat{X}_{0,n} - \frac{\hat{X}_{17,n} + \hat{X}_{18,n}}{2} \right) \). However, the latter step does not guarantee that the new X and Y axes will determine an orthonormal base; therefore, we apply the Gram-Schmidt orthogonalization method so that the hand frame's X and Y axes, say \( \overline{x}_{\text{hand}} \) and \( \overline{y}_{\text{hand}} \), are orthonormal. Finally, the new Z axis is simply obtained as \( \overline{x}_{\text{hand}} \times \overline{y}_{\text{hand}} \) so that the set \( \{ \overline{x}_{\text{hand}}, \overline{y}_{\text{hand}}, \overline{x}_{\text{hand}} \times \overline{y}_{\text{hand}} \} \) is an orthonormal base for the hand's coordinate system. Using this coordinate frame, we can now compute the matrix of change of base that will convey us from the canonical coordinate frame into the hand's coordinate frame. In fact, we now obtain

\[
[T]_{\text{c,hand}} = \begin{bmatrix} \overline{x}_{\text{hand}} & \overline{y}_{\text{hand}} & \overline{x}_{\text{hand}} \times \overline{y}_{\text{hand}} \end{bmatrix},
\]

where \( \mathbf{C} \) denotes the canonical orthonormal base \( \{(1,0,0)^T,(0,1,0)^T,(0,0,1)^T\} \). Thus, we can now obtain the set of 3D coordinates in the canonical base, say \( \{ \mathbf{x}_{i,n} = [T]_{\text{c,hand}}^{-1} \hat{X}_{i,n} \}_{i=0,...,18} \). Next, the conditions are set to allow us determine the directional cosine of each bone and thus model the pose of the hand according to the 3D information available. As attested to by the discussion in the previous section, and in view of the fact that the rotation matrices applied to the
bones are around the origin, in order to compute the directional cosines of each bone, we must first translate the base of the bone under observation so that it coincides with the origin of the global coordinate system. This procedure is repeated for each of the phalanxes; and the transformations, at each step, are accrued in the corresponding accumulated transformation matrix, as shown in (6.1).

As a result of this procedure being repeated for each bone, the last accumulated transformation matrix contains all the information essential to model the hand in its new state.
CHAPTER 7

PRACTICUM AND EXPERIMENTAL RESULTS

As stated before, prior to the 3D reconstruction phase, we must extract the image coordinates of the centroids of the markers on the glove. The tasks involved in achieving this goal were pertinently described in Chapter 4 and are hereby illustrated by Figures 7.1 through 7.3. In fact, Figure 7.1 displays the input images from both cameras, while Figure 7.2 shows the output of the DFS sweep implementing the aforementioned color identification algorithm. Finally, in Figure 7.3, we can observe the accuracy with which the image coordinates of the markers’ centroids have been computed based on the data gathered by the DFS.

We wish to point out the precision of our marker detection algorithm. Indeed, without considering the cases in which occlusion occurs (this research is still in progress), the marker detection succeeds at detecting each and every one of the markers at each step of the tracking phase. Its accuracy relies heavily on the combination of the color identification criteria and the localization of a strong candidate for each marker (Chapter 4).
Figure 7.1: Sample input images from (a) Camera 1 and (b) Camera 2.

Figure 7.2: Results of the DFS pass (detected pixels marked in green) for the pair of input images in Figure 7.1.
In order to test the accuracy of our reconstruction algorithm, we designed a grid with control points. The set contains over 300 control points, positioned every one inch in a grid pattern. In order to conduct the testing, we acquired snapshots of the control grid from each (calibrated) camera. By means of our point extraction algorithm [6], we obtained the image coordinates of all control points within the viewing area common to both cameras (usually a sub-grid of dimensions 21 by 14), and proceeded with the reconstruction procedure as detailed in Chapter 5. Once the reconstruction was complete, we assessed the error between the actual X, Y, and Z coordinates of the control points and their respective reconstructed coordinates. To gauge the accuracy of our algorithm over all the vector coordinates, we consider the reconstruction
error to be the magnitude of the error vector. In Table 1, we present the results obtained with both reconstruction approaches (Sections 5.2 and 5.3) throughout ten trials.

Table 1: Results of the two different approaches of our 3D reconstruction algorithm (Chapter 5) applied to the control grid, with the average taken over 10 different trials.

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>STRAIGHTFORWARD APPROACH</th>
<th>EPIPOLAR APPROACH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>0.111</td>
<td>0.162</td>
</tr>
<tr>
<td>2</td>
<td>0.197</td>
<td>0.159</td>
</tr>
<tr>
<td>3</td>
<td>0.153</td>
<td>0.138</td>
</tr>
<tr>
<td>4</td>
<td>0.098</td>
<td>0.102</td>
</tr>
<tr>
<td>5</td>
<td>0.132</td>
<td>0.172</td>
</tr>
<tr>
<td>6</td>
<td>0.191</td>
<td>0.201</td>
</tr>
<tr>
<td>7</td>
<td>0.109</td>
<td>0.169</td>
</tr>
<tr>
<td>8&lt;sup&gt;6&lt;/sup&gt;</td>
<td>0.106</td>
<td>0.257</td>
</tr>
<tr>
<td>9&lt;sup&gt;6&lt;/sup&gt;</td>
<td>0.201</td>
<td>0.230</td>
</tr>
<tr>
<td>10&lt;sup&gt;6&lt;/sup&gt;</td>
<td>0.181</td>
<td>0.293</td>
</tr>
<tr>
<td>Average Errors</td>
<td>0.1479</td>
<td>0.1883</td>
</tr>
</tbody>
</table>

From Table 1, we can observe that the reconstruction error is fairly low; moreover, it has proven to be uniform throughout our latest experiments.

<sup>6</sup>This trial was carried out with the cameras positioned closer together which resulted in the projection rays forming a smaller angle.
The fact that the reconstruction algorithm implementing the epipolar approach (Section 5.3) did not significantly improve the reconstruction error obtained with the straightforward approach (Section 5.2) can be attributed to the fact that the camera parameters (obtained in the calibration stage) are not exact [6]. Given that we are solving for the point correspondence problem by means of the camera parameters estimated at the calibration stage, the error present in the parameters propagates through the calculations in the point correspondence solution and the reconstruction phase. Hence, in view of the computational cost of the epipolar approach for the reconstruction phase and the slight difference in the performance of both reconstruction algorithms, we have decided to utilize the straightforward approach presented in Section 5.2 for 3D reconstruction purposes.

We wish to note that some of the possible sources of error in our calibration process are the measurement error and the feature extraction, as well as lens distortions. Indeed, the feature extraction accuracy can be affected by non-uniform illumination which causes the detected centroids of the control points to be slightly off the actual center. In regards to the lens distortion, and in view of the fact that our calibration model only considers radial distortion, considering other forms of distortion such as tangential and thin prism may well improve the calibration results [6]. However, there are other issues to ponder while scrutinizing the reconstruction error. As a matter of fact, the
triangulation stage contributes to the overall error since the accuracy of triangulation depends on the angle between the projection rays, as illustrated in Figure 7.4. Thus, the more parallel the projection rays tend to be, the larger the reconstruction uncertainty region becomes.

Figure 7.4: The shaded regions represent the reconstruction uncertainty region which grows larger as the angle between the projection rays becomes smaller.
CHAPTER 8

CONCLUSION

In this thesis, we have presented an algorithm for hand feature estimation by means of marker color detection and reconstruction of three-dimensional Cartesian coordinates. By means of some computationally inexpensive techniques, we were able to efficiently and accurately track the markers on the user's hand and compute the image coordinates of the centroids of all the markers on the glove at every frame (30 fps). Furthermore, the results in Chapter 7 show that our reconstruction algorithm performs well despite several potential sources of error. Hence, our color detection and 3D reconstruction algorithms have proven to be quite robust and precise, allowing us to accurately model the hand via our realistic 3D hand model from the reconstructed coordinates of the centroids of the markers. Altogether, we have presented an approach to modeling a hand's movements. As our research progresses, we hope to more completely approximate the hand's motion by solving for occlusion; we also strive to further reduce the errors incurred at each stage of our project. We are confident that this work not only delivers what has been demanded by the rigid
specifications of our project, but it also presents the potential for several exciting applications in the field of Computer Vision.
APPENDIX A

EPIPOLAR GEOMETRY

A.1 Homogeneous Coordinates

Given a point \((x,y)\) in the plane, its homogeneous coordinates \([x_1, x_2, x_3]\) are any three numbers for which the following equations hold:

\[
\frac{x_1}{x_3} = x \quad \text{and} \quad \frac{x_2}{x_3} = y.
\]

In order to see the connection between the point \((x_1, x_2, x_3) \in \mathbb{R}^3\) and the point \((x,y)\) in the plane with homogeneous coordinates \([x_1, x_2, x_3]\), let us consider the plane \(Z=1\) in space. Thus, the point \((x_1, x_2, x_3)\) in space can be connected to the origin by a line intersecting the plane \(Z=1\) at the point with plane coordinates \((\frac{x_1}{x_3}, \frac{x_2}{x_3})\), or homogeneous coordinates \([x_1, x_2, x_3]\).

It is important to note that homogeneous coordinates of the form \((x_1, x_2, 0)\) for which \(\frac{x_2}{x_1} = \lambda\), describe the point at infinity in the direction of slope \(\lambda\).

The point at infinity can be thought of as the point of intersection of two parallel lines.
Analogously, one can extend the definition of homogeneous coordinates to spaces of higher dimensions. In this thesis, we will consider the homogeneous coordinates of the three-dimensional point \((X,Y,Z)^T\) to be \([X,Y,Z,1]^T\).

### A.2 Right-Pseudo Inverse

Since we are given a matrix \(A_{n \times m}\), with \(n \leq m\), and full row rank \(n\), the system \(Ax = b\) is under-constrained. Therefore, its solution space lies in a subspace \(S \subseteq \mathbb{R}^m\) of dimension \((m-n)\). Hence, in order to make the problem well-posed, we introduce a regularizer (i.e., a cost function that we want to minimize). One obvious choice of a regularizer would be to look for the solution to the given system that is closest to the origin. In that case, the problem of solving \(Ax = b\) can be restated as:

"Find \(\min_{x \in \mathbb{R}^m} \|x\|_2^2\) subject to \(Ax = b\)."

We will show that the solution to (A.2.1), and thus to \(Ax = b\), is given by

\[
x = A^T (A A^T)^{-1} b = A^*_b.
\]

**Proof:**

First, let \(A = U_{n \times n} \left[ \Sigma_n \, \Theta_{n \times (m-n)} \right] V_{m \times m}^T\) be the Singular Value Decomposition (SVD) of \(A\) [19]. Thus, \(U\) and \(V\) are square, orthogonal matrices, i.e.,

\[
U U^T = U^T U = I_n, \quad V V^T = V^T V = I_m, \quad \text{and} \quad \Sigma_n = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) \quad \text{with} \quad \sigma_1 \geq \ldots \geq \sigma_n \quad \text{and} \quad \sigma_i \neq 0, \ 1 \leq i \leq n \quad (\text{for the row-rank of a matrix is equal to the number of non-zero singular values and } A \text{ has row-rank } n). \text{ Then,}
\]
\[ AA^T = U \begin{bmatrix} \Sigma_n & 0 \end{bmatrix} V^T V \begin{bmatrix} \Sigma_n \end{bmatrix} U^T = U \begin{bmatrix} \Sigma_n & 0 \end{bmatrix} I_n \begin{bmatrix} \Sigma_n \end{bmatrix} U^T \]  

(A.2.3)

where \( \Sigma_n = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2) \). Therefore, \( AA^T \) is invertible; furthermore, \( AA^T \) is symmetric and positive-definite.

In order to solve (A.2.1), we can use Lagrange multipliers. Hence, we introduce a new variable \( \lambda \) (the Lagrange multiplier), and consider the function

\[ h(x, \lambda) = \|x\|_2^2 + \lambda^T (Ax - b) = x^T x + \lambda^T (Ax - b). \]  

(A.2.4)

The points at which \( \|x\|_2^2 = x^T x \) is minimized are contained in the solution set of the equation \( \frac{\partial h}{\partial x} = 0 \) along with \( Ax = b \). Thus, since \( \frac{\partial h}{\partial x} = 2x^T + \lambda^T A \), it follows that

\[ 2x^T + \lambda^T A = 0 \]

\[ x^T = \frac{\lambda^T}{2} A. \]  

(A.2.5)

\[ x = A^T \frac{\lambda}{2} \]

Thus, substituting \( x \) in \( Ax = b \), we obtain that \( b = A \left( A^T \frac{\lambda}{2} \right) = AA^T \frac{\lambda}{2} \). Then, given that \( AA^T \) is invertible, it follows that \( \lambda = 2 \left( AA^T \right)^{-1} b \) so that (A.2.2) results.
It is important to note that $A^T_i = A^T (AA^T)^{-1}$ is called the right-pseudo inverse because $AA^T_i = AA^T (AA^T)^{-1} = I$. In general, however, $A^T_i A \neq I$. In fact, $A^T_i A = I$ iff $A$ is square and invertible, in which case $A^T_i = A^{-1}$.

A.3 Definition of $[y]_x$

Given $y = (y_1, y_2, y_3)^T$, the matrix $[y]_x$ is defined by

$$[y]_x = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix}.$$ 

Clearly, $[y]_x$ is skew-symmetric - in fact, $a_{ij} = -a_{ji}$ for all $1 \leq i, j \leq 3$ - and singular. Moreover, if $z = (z_1, z_2, z_3)^T$ is another 3D vector, it follows that

$$y \times z = [y]_x z,$$

where $\times$ represents the cross product operator.

Therefore, referring back to (3.10), since $\ell_{12} = (P^2 \mathbf{C}_1) \times [P^2 \left(\begin{array}{c} P^1 \\ \hat{p}_1 \end{array}\right)]$ and $P^2 \mathbf{C}_1 = \mathbf{e}_2$, one obtains

$$\ell_{12} = (P^2 \mathbf{C}_1) \times \left[ P^2 \left(\begin{array}{c} P^1 \\ \hat{p}_1 \end{array}\right) \right] = \mathbf{e}_2 \times \left( P^2 \left(\begin{array}{c} P^1 \\ \hat{p}_1 \end{array}\right) \right) = [\mathbf{e}_2]_x \left( P^2 \left(\begin{array}{c} P^1 \\ \hat{p}_1 \end{array}\right) \right).$$
A.4 Derivation of the Fundamental Matrix in Terms of the Projection Matrices \( P^1 \) and \( P^2 \)

By (3.13), we know that \( F = \begin{bmatrix} [\epsilon_2], P^2 (P^1)^\top \end{bmatrix} \). Given that the global origin is set to be at the optical center of camera 1, \( C_1 \), and \( \epsilon_2 = P^2 C_1 \), we obtain that

\[
\epsilon_2 = P^2 C_1 = K^2 \begin{bmatrix} R \mid t \end{bmatrix} \begin{bmatrix} 0_{3 \times 3} \mid 1 \end{bmatrix} = K^2 t. \tag{A.4.1}
\]

Then,

\[
F = \begin{bmatrix} K^2 t \end{bmatrix}, P^2 (P^1)^\top = \begin{bmatrix} K^2 t \end{bmatrix}, K^2 \begin{bmatrix} R \mid t \end{bmatrix} (P^1)^\top. \tag{A.4.2}
\]

However, by algebraic manipulation, it can be shown that for any matrix \( A \),

\[(Ax) \times (Ay) = kA^\top (x \times y),\]

for all \( x, y \) and for some scalar \( k \). In particular, since the above equality is valid for all \( y \), it follows that

\[\begin{bmatrix} Ax \end{bmatrix} A = kA^\top [x].\]

Thus, returning to (A.4.2), and recalling that the effects of the fundamental matrix are not affected by scalar factors, we have that

\[
F = (K^2)^{\top} \begin{bmatrix} t \end{bmatrix}, \begin{bmatrix} R \mid t \end{bmatrix} (P^1)^\top. \tag{A.4.3}
\]

Moreover, from Appendix B.2, we know that \( (P^1)^\top = (P^1)^\top \left[ P^1 (P^1)^\top \right]^{-1} \), so that

65
\[ (P^1)^T = (P^1)^T [P^1 (P^1)^T]^{-1} \]
\[
= \begin{bmatrix}
I_3 \\
o_{1\times 3}
\end{bmatrix}
(K^1)^T
\begin{bmatrix}
K^1 & [I_3 | o_{3\times 1}]
\end{bmatrix}
\begin{bmatrix}
I_3 \\
o_{1\times 3}
\end{bmatrix}
(K^1)^T
\]
\[
= \begin{bmatrix}
I_3 \\
o_{1\times 3}
\end{bmatrix}
(K^1)^T
\begin{bmatrix}
K^1 & I_3 (K^1)^T
\end{bmatrix}^{-1}
\]
\[
= \begin{bmatrix}
I_3 \\
o_{1\times 3}
\end{bmatrix}
(K^1)^T
(K^1)^{-T}
(K^1)^{-1}
\]
\[
= \begin{bmatrix}
(K^1)^{-1T} \\
o_{1\times 3}
\end{bmatrix}
. \quad (A.4.4)
\]

Then,
\[
F = (K^2)^{-T} [t], [R \mid t]
\begin{bmatrix}
(K^1)^{-1T} \\
o_{1\times 3}
\end{bmatrix}
\]
\[
= (K^2)^{-T} [t], [R \mid t]
\begin{bmatrix}
(K^1)^{-1T} \\
o_{1\times 3}
\end{bmatrix}
\]
\[
= (K^2)^{-T} [t], R (K^1)^{-1} . \quad (A.4.5)
\]
APPENDIX B

IMAGE PROCESSING

B.1 Image Equalization

An image can be characterized based on statistics derived from the entire set of constituent pixels. The set of descriptors thus obtained is referred to as global features of the image. Usually, these features are obtained via the study of an intensity histogram of the image [10, 11, 12].

The image intensity histogram is a plot that shows the number of image pixels that display each of the possible discrete intensity values. Therefore, in order to construct an intensity histogram, each constituent pixel is examined so that its intensity value can be determined. Then, we count the number of pixels displaying each of the possible values. When plotting the histogram, each intensity value is represented by a histogram bin whose weight represents the number of pixels displaying that intensity. Since we are working with RGB images, the histogram is obtained by separately computing an intensity histogram for each of the monochrome bands (i.e., the red, green and blue bands), as shown in Figure B.1.
By using a histogram, one can extract valuable information regarding the global features of the image. Indeed, one of the very useful features of a histogram is the range of intensity levels, which is defined as the difference between the maximum and the minimum pixel values represented in the histogram and referred to as the *Dynamic Range*. This information can be used, among other things, to enhance the quality of the image, for example, by increasing the visual contrast of an image in a designated intensity range or ranges. In fact, the histogram of the image can be transformed so that the new histogram reflects the changes in contrast in the range of interest.
Definition:

We define the contrast value \( C(x, y) \) at pixel \((x, y)\) as the difference between its intensity value \( I(x, y) \) and the average background intensity \( \bar{I} \) normalized by the full intensity range; that is,

\[
C(x, y) = \frac{I(x, y) - \bar{I}}{I_{\text{max}}}.\]

Thus, in order to enhance the contrast of the image, we must stretch the occupied intensity range.

Given that our goal is to facilitate the color detection by enhancing the contrast of the image, a straightforward approach to the histogram transformation is needed. For that reason, the so-called Histogram Expansion transformation was chosen [11]. This is the most straightforward and conservative of the transformations, and it basically stretches the nonzero input intensity range, \([x_{\text{min}}, x_{\text{max}}]\), to an output intensity range \([0, y_{\text{max}}]\). Hence, each intensity value \( x \) is mapped onto the new range by means of the following linear mapping:

\[
y = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} y_{\text{max}}.
\]

This mapping increases the contrast without modifying the shape of the original histogram. However, sometimes the lowest and highest occupied histogram intensity bins can be close to the minimum and the maximum of the full range, yet the image has low contrast. In such a case, a simple Histogram Expansion will not prove effective. Nevertheless, a
suitable modification of the expansion can be introduced in order to overcome this difficulty. In fact, it is sufficient to include a user-defined cut-off value $p$ so that $p\%$ of the range is then expanded over the dynamic range. In the implementation of this technique, $x_{min}$ and $x_{max}$ are not the actual minimum and maximum intensity values present in the original image. Instead, we have new values $x'_{min}$ and $x'_{max}$ that are determined according to the following formulae:

$$
\begin{align*}
x'_{min} &= x_{min} + (1 - p \times 0.01) \times \frac{\text{Dynamic Range}}{2}, \\
x'_{max} &= x_{max} - (1 - p \times 0.01) \times \frac{\text{Dynamic Range}}{2},
\end{align*}
$$

where \textit{Dynamic Range} is the dynamic range of the original image. The effects of the Histogram Expansion on the image in Figure B.1 (a) can be seen in Figures B.2 and B.3 where different cutoff values were chosen.
Figure B.2: (a). Processed image ($p = 85$). (b). Expanded histograms of the modified image.

Figure B.3: (a). Processed image ($p = 75$). (b). Expanded histograms of the modified image.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
It is clear that after the Histogram Expansion, the colors on the markers appear sharper and richer than in the original image which, in turn, clearly improves the performance of the color identification algorithm.

B.2 Criteria for the Selection of Strong Candidates

Given that in the initialization of the tracking phase the search for the markers is performed without any type of information as to their whereabouts, we must ensure that we can successfully detect the constituent pixels of each marker. Thus, when a pixel has been detected to be of one of the colors of interest (red or green), we do not ensue with the local search process unless we are certain that the particular pixel has the potential to belong to a marker. Hence, we inspect all the immediate pixels surrounding it (Figure B.2.1); and only when all of those are of the same color as the pixel detected, we label it as a strong candidate.

Figure B.4: The blue dots represent the neighboring pixels that are inspected in order to determine whether the pixel of interest (black dot) is a strong candidate during the initialization of the tracking phase.

On the other hand, given that in the tracking phase the detection of the markers is based on previous knowledge of their location (from
previous frames), we can relax the requirements for a pixel to be a strong candidate. In fact, the small discrepancy between frames along with our color identification algorithm allows us to accurately predict the location of the marker in the frame being processed.

In the worst-case scenario, the predicted point will lie on the edge of the marker. This situation could not have been allowed in the initialization phase given that we lacked information (at that stage) to assure us of the suitability of the pixel. However, in the tracking phase, we are certain that the pixel lies on a marker. Yet, we guarantee it by inspecting some of its surrounding pixels. As mentioned before, since it is possible to have the predicted point lie on the edge of the marker, it will not be feasible to require the pixel to be surrounded by eight pixels of its same color as in Figure B.4.; thus the condition we impose is that it is surrounded by at least four pixels of its same color as depicted in Figure B.5. In fact, the pixel under observation is labeled as a strong candidate if three of the four pixels required lie to only one of the sides of the dotted line (blue dots), whereas the remaining is any (or both) of the pixels lying at the ends of the dividing segment (orange dots).
Figure B.5: Cases considered when inspecting a potential strong candidate (black dot) during the tracking phase.
A rotation in three dimensions can be viewed as a rotation around the three main axes, that is, the X, Y and Z axes. When considering one of these main axes to be the rotation axis, the problem resembles the two dimensional case. Thus, when considering the rotation around the X axis, only the Y and Z coordinates need to be updated (see Figure C.1).

Figure C.1: Rotation about the X axis.
In fact, from Figure C.1 it follows that

\[
\begin{align*}
  x' &= x \\
  y' &= \cos(\alpha + \gamma), \\
  z' &= \sin(\alpha + \gamma)
\end{align*}
\]  
(C.1)

and

\[
\begin{align*}
  x &= x \\
  y &= \cos(\gamma), \\
  z &= \sin(\gamma)
\end{align*}
\]  
(C.2)

However, from (C.1), we have that

\[
y' = \cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\gamma)
\]  
(C.3)

and

\[
z' = \sin(\alpha)\sin(\gamma) + \cos(\alpha)\cos(\gamma).
\]  
(C.4)

Then, combining (C.2) – (C.4), we obtain

\[
\begin{align*}
x' &= x \\
y' &= y\cdot\cos(\alpha) - z\cdot\sin(\alpha). \\
z' &= z\cdot\cos(\alpha) + y\cdot\sin(\alpha)
\end{align*}
\]  
(C.5)

Thus, the equations in (C.5) can be expressed matricially as follows:

\[
R_x(\alpha) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\alpha) & -\sin(\alpha) & 0 \\
0 & \sin(\alpha) & \cos(\alpha) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  
(C.6)

The matrix in (C.6) is referred to as the matrix of rotation about the X axis by an angle \( \alpha \). Note that \( R_x(\alpha) \) is expressed using homogeneous
coordinates (Appendix B.1), for this type of representation is of extreme
importance when working with the OpenGL graphic engine.

In a similar fashion, one can derive the respective matrices of rotation
about the remaining axes, namely,

\[
R_y(\beta) = \begin{bmatrix}
\cos(\beta) & 0 & -\sin(\beta) & 0 \\
\sin(\beta) & 1 & \cos(\beta) & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}, \quad (C.7)
\]

and

\[
R_z(\phi) = \begin{bmatrix}
\cos(\phi) & -\sin(\phi) & 0 & 0 \\
\sin(\phi) & \cos(\phi) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}. \quad (C.8)
\]

Thus, if our goal is to perform a rotation of \( \phi_x \) degrees around the X axis
followed by a rotation of \( \phi_y \) degrees around the Y axis, and finally rotate
it all by \( \phi_z \) degrees around the Z axis, then our total rotation matrix is
given by

\[
R(\phi_x, \phi_y, \phi_z) = R_z(\phi_z) \cdot R_y(\phi_y) \cdot R_x(\phi_x), \quad (C.9)
\]

where \( R_x(\phi_x) \), \( R_y(\phi_y) \), and \( R_z(\phi_z) \) are the matrices in (C.6), (C.7) and
(C.8), respectively.

It is crucial to note that the order of the operations in (C.9) is of
extreme relevance. In fact, the order of the factors will significantly
change the outcome. Therefore, one must know the order of the
rotations to perform prior to computing the total rotation matrix
\[ R(\phi_x, \phi_y, \phi_z) \] in order to adjust the factors in (C.9).
REFERENCES


VITA

Graduate College
University Of Nevada, Las Vegas

Maria Beatriz Teruel

Home Address:
3875 Cambridge St. Apt. 216
Las Vegas, NV 89119

Degrees:
Bachelor of Science, Mathematical Sciences 1999
National Southern University, Argentina

Master of Science, Mathematical Sciences 2003
University of Nevada, Las Vegas

Publications:


Thesis Title: Color Detection and 3D Reconstruction for Hand Pose Estimation as Part of an HCI System

Thesis Examination Committee:
Chairperson, Evangelos Yfantis, Ph. D.
Committee Member, Dr. Wolfgang Bein, Ph.D
Committee Member, Dr. Ajoy Datta, Ph.D
Graduate Faculty Representative, Dr. David Costa, Ph.D.