AUTOMATED CALIBRATION AND REGISTRATION USING ACTIVE APPEARANCE MODELS FOR A FINGERNAIL IMAGING SYSTEM

by

Thomas R. Grieve

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering

The University of Utah

August 2014

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Thomas R. Grieve**

has been approved by the following supervisory committee members:

Stephen A. Mascaro	, Chair	**25 July 2013** Date Approved
John M. Hollerbach	, Member	**25 July 2013** Date Approved
Mark A. Minor	, Member	**N/A** Date Approved
Stacy M. Bamberg	, Member	**30 July 2013** Date Approved
Yu Sun	, Member	**25 July 2013** Date Approved

and by **Tim A. Ameel**, Chair/Dean of

the Department/College/School of **Mechanical Engineering**

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Fingernail imaging is a method of sensing finger force using the color patterns on the nail and surrounding skin. These patterns form as the underlying tissue is compressed and blood pools in the surrounding vessels. Photos of the finger and surrounding skin may be correlated to the magnitude and direction of force on the fingerpad.

An automated calibration routine is developed to improve the data-collection process. This includes a novel hybrid force/position controller that manages the interaction between the fingerpad and a flat surface, implemented on a Magnetic Levitation Haptic Device. The kinematic and dynamics parameters of the system are characterized in order to appropriately design a nonlinear compensator. The controller settles within 0.13 s with less than 30% overshoot.

A new registration technique, based on Active Appearance Models, is presented. Since this method accounts for the variation inherent in the finger, it reduces registration and force prediction errors while removing the need to tune registration parameters or reject unregistered images. Modifications to the standard model are also investigated. The number of landmark points is reduced to 25 points with no loss of accuracy, while the use of the green channel is found to have no significant effect on either registration or force prediction accuracy.

Several force prediction models are characterized, and the EigenNail Magnitude Model, a Principal Component Regression model on the gray-level intensity, is shown to fit the data most accurately. The mean force prediction error using this prediction and modeling method is 0.55 N. White LEDs and green LEDs are shown to have no statistically significant effect on registration or force prediction. Finally, two different calibration grid designs are compared and found to have no significant effect.

Together, these improvements prepare the way for fingernail imaging to be used in less controlled situations. With a wider range of calibration data and a more robust registration method, a larger range of force data may be predicted. Potential applications for this technology include human-computer interaction and measuring finger interaction forces during grasping experiments.

For my wife, Lisa, whose love and support have been invaluable, and my daughter, Alice, a dream who has become a reality.

CONTENTS

ABSTRACT iii

LIST OF TABLES viii

CHAPTERS

1. INTRODUCTION 1

1.1 Interacting with the Human Fingerpad 2
1.1.1 Force Sensing 4
1.1.2 Force Control 6
1.2 Image Registration 7
1.3 Modeling and Optimization 10
1.3.1 Force Prediction Models 11
1.3.2 Lighting 11
1.4 Summary 13

2. 3-D FINGERTIP TOUCH FORCE PREDICTION USING FINGERNAIL IMAGING WITH AUTOMATED CALIBRATION . . . 15

2.1 Abstract 15
2.2 Introduction 15
2.2.1 Background 16
2.2.2 Current Work 17
2.3 Calibration Procedures 18
2.3.1 Experimental Apparatus 18
2.3.2 Interface Design 21
2.3.3 Experiment Design 22
2.3.4 Image Registration 22
2.4 Control Design 23
2.4.1 Controller Requirements 23
2.4.2 System Model 24
2.4.3 System Identification 25
2.4.4 Controller Design 27
2.5 Results 29
2.5.1 Controller Implementation 30
2.5.2 Trajectory Tracking 33
2.5.3 Model Calibration - Force Prediction 37
2.5.4 Calibration Grids 39
2.6 Conclusion 40

3. IMAGE REGISTRATION USING ACTIVE APPEARANCE MODELS FOR CALIBRATION OF FINGERNAIL IMAGING 42
3.1 Abstract 42
3.2 Introduction 42
3.2.1 Prior Registration Methods 44
3.2.2 Active Appearance Models 45
3.2.3 Current Work 47
3.3 Experimental Procedures 48
3.4 Registration Procedure 48
3.4.1 Training Images 49
3.4.2 Landmark Points 50
3.4.3 Shape Model 56
3.4.4 Texture Model 57
3.4.5 Appearance Model 58
3.4.6 Search Model 60
3.5 Registration Error 61
3.6 Method Modifications 62
3.6.1 Landmark Point Density 62
3.6.2 Color Channel Registration Models 64
3.6.3 Modified Shape Registration Model 65
3.7 Force Prediction 67
3.8 Conclusion 70

4. OPTIMIZING FINGERNAIL IMAGING CALIBRATION FOR FORCE PREDICTION 72
4.1 Abstract 72
4.2 Introduction 72
4.3 Calibration Setup 74
4.3.1 Experimental Apparatus 74
4.3.2 Calibration Grids 75
4.3.3 Experiment Design 77
4.3.4 Image Registration 78
4.4 Force Prediction Models 78
4.4.1 Linearized Sigmoid Model 78
4.4.2 EigenNail Magnitude Model 80
4.4.3 AAM Parameters Models 81
4.5 Results 82
4.5.1 Lighting 82
4.5.2 Calibration Grids 83
4.5.3 Force Prediction Models 87
4.6 Conclusion 92

5. CONCLUSION 93
5.1 Future Work 94

APPENDICES

A. ADDITIONAL CONTROLLER DETAILS 96

B. ADDITIONAL REGISTRATION DETAILS 109

C. ADDITIONAL MODELING AND OPTIMIZATION DETAILS 123

REFERENCES 126

LIST OF TABLES

2.1 Flotor Inertial Parameters 27
2.2 Controller Gains 30
A.1 Flotor/Camera Static Parameters 105
A.2 Observability of Calibration and Validation Trajectories 106
C.1 Grid Choice and Region Order for Each Data Set 125

CHAPTER 1

INTRODUCTION

Fingernail imaging is a method of sensing fingerpad force using the color patterns created by the flow of blood through the tissue beneath the nail and surrounding skin [1]. These color patterns are formed as blood vessels within the finger are compressed by the interaction between the distal phalanx bone, the fingernail and the soft tissue surrounding the blood vessels. As the tissue compresses, blood collects and skin reddens in certain regions while blood evacuates and skin whitens in other regions. The resulting patterns correlate with the magnitude and direction of force on the fingerpad, indicating that they may be used as a general method of transducing finger forces [2].

Mascaro and Asada developed the original fingernail sensors, which consisted of an array of photodetectors and infrared light-emitting diodes (LEDs) mounted on a printed circuit board (PCB), with optically transparent epoxy to attach the board to the finger and optically opaque epoxy to attach the back side of the board to a hard plastic nail [1]. These fingernail sensors were custom-manufactured for each test subject, with 6 infrared LEDs and 8 photodetectors arranged to cover the region of the nail displaying the most visible color changes. The sensors were tested on 16 subjects using a manual calibration routine, with a spiraling trajectory rather than specific targets in the force space. The trajectory guided the subjects to apply normal forces up to 4 N and shear forces up to 2 N. The calibration process resulted in a root-mean-squared (RMS) error on validation data of 1.4 N normal force and 0.6 N shear force.

This technique was expanded to use a digital camera with an external light source for detecting the color patterns [3], essentially increasing the number of photodetectors to equal the number of pixels in the image while eliminating the need for custom manufacturing of sensors. In addition, the surrounding skin is included in the imaging to benefit from the color changes that occur there. The tradeoffs, however, are the need to control the lighting and the requirement that the finger remain in view of the camera at all times. Expanding the regions of the nail and skin that are imaged allows a larger range of forces to be transduced. The original fingernail imaging experiments involved normal forces up to 10 N

and shear forces up to 3 N. Modeling and prediction were performed on one direction of force at a time. In other words, only images involving z-direction force were used to form and validate the z-direction model, and only images involving y-direction force (and the normal force required to maintain frictional contact) were used to form and validate the y-direction model. This method resulted in validation RMS errors of 0.3 N normal force and 0.6 N shear force.

Some important results have come from attempts at optimizing and refining both techniques. When using fingernail imaging, it was determined that the green channel of an RGB image correlated most closely with the forces [4]. In addition, the wavelength of light emitted by the LEDs in fingernail sensors was studied to determine an optimal LED color. Green LEDs, having a wavelength of 525 nm, were found to provide the best force response [5]. The requirement for fingernail sensors to be custom-manufactured has been eliminated by the introduction of a flexible PCB [6]. Using data from fingernail imaging, it was found that the photodetectors and LEDs in fingernail sensors are optimally placed when they are as near as possible [5].

This work addresses improvements made to the fingernail imaging process, including the creation of an automated system for recording calibration data and the development of a new registration process as well as detailing the optimization of the modeling and lighting used during fingernail imaging calibration.

1.1 Interacting with the Human Fingerpad

Two primary concerns related to interacting with the human fingerpad are relevant to this work. First, sensing the contact forces (normal and shear) between the fingerpad and an object has many applications and has been performed in a large number of ways. All other available methods restrict the finger in some way, however. Camera-based fingernail imaging removes these restrictions. Second, three-dimensional force control of a robot interacting with the fingerpad, using a flat surface as the end effector, has not been studied. As such control is required to develop an automatic calibration system, it is important to the improvement of fingernail imaging systems.

First, some definitions regarding the hand are given (Fig. 1.1). The joints of each finger (index, middle, ring and little) are named, beginning with the knuckle, the metacarpophalangeal (MCP) joint, the proximal interphalangeal (PIP) joint, and the distal interphalangeal (DIP) joint. The bones of each of the fingers, beginning at the MCP, are named the proximal phalanx, the middle phalanx and the distal phalanx. The joints of the thumb are, beginning with the knuckle, the metacarpophalangeal (MCP) joint and the interphalangeal

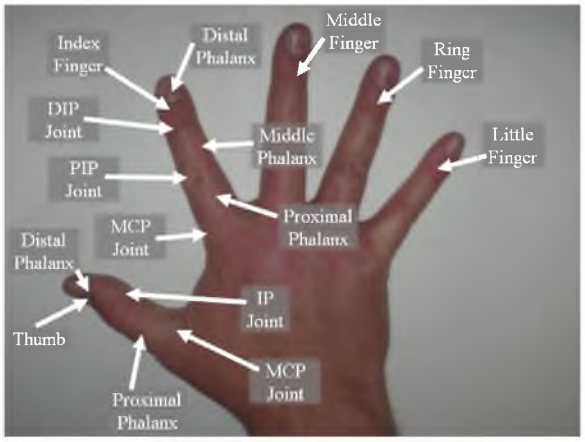

Fig. 1.1. Parts of the human hand. The joints and bones of the index finger and thumb are labeled. Although this work is concerned only with the thumb and index finger of the right hand, the joints and bones of the other three fingers are named identically to those of the index finger.

(IP) joint. The bones of the thumb, beginning at the MCP, are the proximal phalanx and the distal phalanx. This work is concerned only with the thumb and index finger of the right hand.

Some conventions are established concerning force directions on the human fingerpad. A right-handed coordinate system is used, where the z-axis points vertically away from the nail and the y-axis points distally. When the finger pushes in the direction of the positive axis, the force is positive in that direction, so when the finger pushes to the right (as viewed from the perspective of Fig. 1.2), a positive x-direction force is generated. When the finger presses normally against a flat surface, a negative z-direction force is generated. Alternatively, when the finger is stationary and a platform pushes up (as viewed from the perspective of Fig. 1.2), a negative y-direction force is generated since it is the same as the

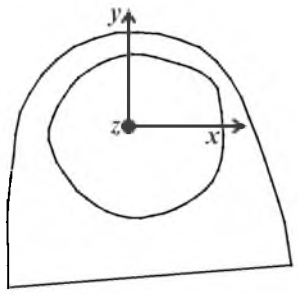

Fig. 1.2. Force conventions on the finger. A right-handed coordinate system is used, where the z-axis points vertically away from the nail and the y-axis points distally. When the finger pushes in the direction of the positive axis, the force is positive in that direction.

finger pushing down. Using planar contact without adhesive, it is not possible for the finger to exert a positive z-direction force.

1.1.1 Force Sensing

Fingernail imaging differs from other methods of fingerpad force sensing in that it permits the contact force to be measured without restricting the haptic senses and without requiring that force sensors be precisely placed in prespecified contact locations. Using fingernail imaging to measure precision grasping force would simplify many grasping studies since it would remove these restrictive requirements.

Many researchers have studied human precision grasp force [7–11], as it has several potential applications. A better understanding of the interaction among the fingers during grasping should help with the design of improved prosthetic devices [12]. Understanding the force on the fingers and the frictional properties of skin can help engineers design products that can be grasped more easily [13]. When loss of limb function and sensation occurs, neuromuscular electrical stimulation can be used in rehabilitation [14]. Such stimulation could be improved by a better understanding of the relationship between the central nervous system and the finger forces. Studies have been performed on the effects on grasping of different neurological conditions, such as Parkinson's Disease [15, 16] and stroke [17]. To examine the role of tactile sensation in grasp force control, anesthesia has been applied to the hand in several different experiments [11, 18, 19].

A typical grasping study uses a manipulandum with sensors beneath each finger [20], as illustrated in Fig. 1.3(a). The test subject lifts the object while attempting to minimize the tilt angle. A weight is placed in one of multiple locations (marked A, B, and C) to change the location of the center of mass. The force sensor locations are fixed, and the fingers are restricted to those specific contact points so that the force sensors may measure the finger contact forces. These forces are used to describe the interaction between the fingers as the object is grasped.

However, such a constrained grasp may introduce confounding factors. One experiment compared the effect of Parkinson's Disease (PD) on the ability of subjects to coordinate multifingered grasp forces [15]. It was determined that all subjects (both with and without PD) were able to modulate individual fingertip forces to minimize object tilt. Although medication improved their abilities, subjects with PD who were off medication were demonstrably slower to adjust their finger forces to compensate for the changed center-of-mass location. Since these tests were performed using specified contact locations, however, it might be suggested that the effects of prespecified contact locations are conflated with the

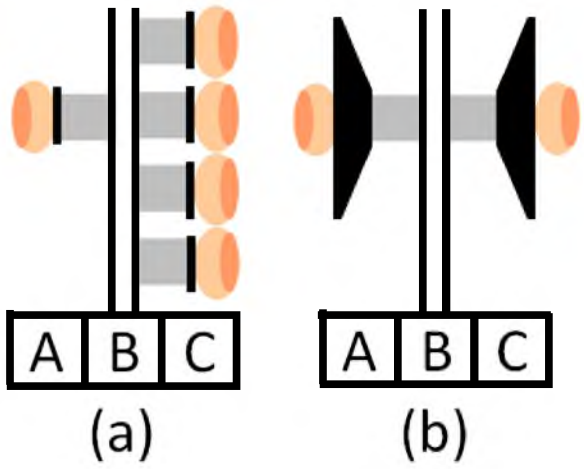

Fig. 1.3. Schematic of typical grasping experiments. (a) Constrained grasping experimental apparatus. Finger contact positions are defined by the locations of the force sensors along the sides of the object to be grasped. (b) Unconstrained grasping apparatus. Fingers may contact anywhere along the pads on both sides. With both objects, the center of mass of the object may be changed by adding a weight in positions A, B, or C.

effects of the task itself. In other words, one might question whether the difficulty lay in forcing the fingers to hold what might be an unnatural posture or in modulating the fingertip forces to adjust for object tilt.

Implementing fingernail imaging in a grasping experiment to measure the individual finger forces would allow each test subject to hold the manipulandum in an individual, natural grip. No prespecified contact locations would be required because no force sensors would be needed. Instead, the finger forces would be measured with a camera on either side of the object, recording the finger coloration patterns, which could then be transformed into finger forces. Additionally, any reasonable object could be used in a grasping experiment. Custom objects would not need to be manufactured for testing. Concerns about different hand sizes would no longer restrict test subjects from participation. The fingers could be placed anywhere on the object and oriented in any direction relative to the object as long as a camera could see the nail and surrounding skin.

Interest in finding a method of performing such an unconstrained grasping test already exists in the field. One study demonstrating this concern investigated the effects of constrained and unconstrained grasping on two-fingered grasps [8]. In addition to a constrained grasping apparatus, a new, unconstrained grasping apparatus was designed, as illustrated in Fig. 1.3(b). Test subjects were required to use the predefined contact points on the constrained apparatus but were free to choose contact locations on the unconstrained object. The results of these experiments demonstrate that lower overall forces are used when contact

locations are not constrained. This is compensated by a high variability in contact position. (It should be noted that the prior study, involving subjects with PD, found that the forces were not as well coordinated in multifingered grasping as in two-fingered grasping [15].) Conversely, in constrained grasping, the forces must be higher while the contact locations are consistent. One conclusion drawn was that the finger force and posture have a complex interrelationship in anticipatory grasping, which is controlled through the central nervous system. As PD affects the central nervous system, it might be suggested that a subject with PD could struggle more with anticipatory control of finger force during constrained grasping than during unconstrained grasping.

Although the last study demonstrates two-fingered unconstrained grasping, the researchers note that it would be difficult to extend the concept to three or more fingers. If fingernail imaging is used to detect the force on the fingers, however, all fingers could be used in either constrained or unconstrained grasping as long as all are visible to a camera. This would allow a researcher to restrict the potential explanations for the outcome of experiments.

1.1.2 Force Control

While many techniques exist for controlling force, research into the application of controlled force to the human fingerpad has been limited. Single-axis actuators have been used to study finger dynamics, such as the stiffness to normal force as a function of contact angle [21] and to determine the pressure distribution along the finger with a constant contact angle [22]. To study friction (or shear) forces, a single-axis actuator was used to apply controlled shear forces in a single direction while the contact angle varied [23]. These single-axis actuators all implemented position rather than force control and so are only tangentially relevant to this work.

A high-performance 3-axis force controller was developed to allow a haptic device to apply 3-axis forces to the fingertip [24]. However, this method placed the fingertip in a thimble, which applies the forces to the finger as a whole rather than to the fingerpad. Thus, the horizontal forces were more like normal force applied along the sides of the finger rather than frictional shear forces applied to the fingerpad.

One other true three-dimensional force controller applying forces between the human fingerpad and a flat surface has been implemented [25]. This controller uses two connected Novint Falcons with an ATI Nano17 6-axis force sensor to apply three-dimensional forces and z-direction torque to the fingerpad. The controller has a settling time of 0.25 s and is able to track desired trajectories with RMS errors of 0.05 N or less in all three force

directions and 0.39 N in z-direction torque. It should be noted that although the paper title claims 5-DOF (degrees-of-freedom) force control, the paper repeatedly acknowledges that the 5th degree of freedom, y-direction torque, does not produce consistent results due to the difficulty of reliably locating the finger with respect to the force sensor. The inability of the system to properly track and control torque in the y-direction makes sense as the sensor is not really detecting a pure torque exerted by the finger on the flat plate, but the effect of the x- and z-directions forces, combined with the appropriate moment arms. In addition, the controller settling time is approximately 2.5 times as long as that of the one designed here. This would greatly increase calibration time while reducing accuracy. One final potential problem is that no maximum force range is mentioned in the paper. The plots show up to 4 N of normal force being applied, but no mention is made of the absolute maximum of which the mechanism is capable. If a much larger range is not possible, the mechanism would not be suitable for this application.

Studies of finger dynamics [21–23] demonstrate that the dynamic behavior of the fingerpad is complex and nonlinear. Applying force control to the fingerpad will result in coupling between the normal and shear forces. The dynamic behavior of the fingerpad and the coupling of forces in the three axes due to finger geometry and frictional constraints make this a unique and challenging problem.

1.2 Image Registration

Finger image registration has been studied extensively for use in predicting finger forces. Prior methods used for fingernail image registration include 2D-to-3D Registration, the Harris Method, the Canny Method, the Scaled Rigid-Body Transform, and the Hand-to-Finger Transform. Each method has its benefits as well as its drawbacks.

The *2D-to-3D registration method* [3] uses fiducial marks drawn on the finger to register each image. First, a stereo image of the finger is used to generate a 3-D model of the finger. Then, 2-D images are registered to this model by relating the location of the fiducial marks in the 2-D images to their coordinates in the 3-D model. While this method has been seen to be accurate, the complexity of the model requires longer processing time. In addition, relevant color information may be obscured by the fiducial marks. All other techniques assume the finger to be a flat plane and attempt to register the finger to some template image.

The *Harris method* [26, 27] correlates the Harris feature points [28] from an image to be registered with feature points in the template image. This correlation is based on pixel intensities in the neighborhood surrounding each pixel. The RANSAC (RANdom

SAmple Consensus) method [29] is then applied to find the transformation that best fits the correlations. This method fails to register some images due to the lack of well-defined feature points on the finger. It also frequently identifies correlations between points that do not correspond, whether due to the changing color patterns on the finger or the similarity of large regions of the finger and nail. These miscorrelations result in skewed images that are improperly registered. Additionally, since the RANSAC algorithm is probabilistic, the method is not repeatable. It has correctly registered an image on one attempt and failed to do so on another, simply because of noise in the image and the random number generator.

The *Canny method* [26] uses Canny edge-finding [30] to locate the edge points in both the template image and a new sample image. As with the Harris Method, these edge points are then correlated between the sample image and the template image, and RANSAC is used to find the correlated pairs that produce a minimum-error transformation. This method has been more successful at registering images than the Harris method, but has proven to be slower. It also requires the optimal values of the parameters (i.e., the high and low thresholds and the standard deviation of the Gaussian) to be experimentally determined for each data set, based on the lighting conditions as well as the shape of the finger itself. In addition, when the proximal end of the finger is obscured, a false edge is found that may provide incorrect registration information to the algorithm.

The *Scaled rigid-body transform* [26, 31] first locates the finger in the image using thresholding, given a predefined skin-color vector. Connected-component (CC) analysis is then used to find all pixels belonging to the largest region in the image, which is assumed to be the finger. The rotational offset of the finger is estimated from the major axis of this CC. The finger is rotated to align with the vertical axis, and the image is cropped to the edge of the CC. The model assumes that the motion of the finger is limited to (1) motion in and out of the image plane (i.e., scaling) and (2) motion parallel to the image plane (i.e., x-y translations and z-rotation). Although this method is fast and accurate for predicting force, it does not accurately register the images. The seemingly accurate results previously demonstrated in force prediction [26] are an effect of the movement of the finger within the bounding-box of the final image, rather than the changing intensity of the finger regions themselves. In other words, the model is calibrated to the movement of the finger within the bounding-box, not simply to the coloration of the finger. This failure is because the basic assumptions of this model fail. While experience has shown that rotations out of the plane (i.e., x-rotation and y-rotation) are negligible, one other modality is ignored: the finger may deform as force is applied, leading to apparent motion between the nail and the

edge of the finger.

The *Hand-to-finger transform* [32] begins with an image of the entire hand. First, the hand is found using a predefined skin color vector. The angle between each pixel intensity and this color vector is calculated, and if the angle is less than a given threshold, the pixel is defined to be part of the hand. The fingertips are found using Integral Image [33] to search for hand pixels whose neighborhood contains more than a given threshold of other hand pixels. The finger direction is estimated by searching for the chord of a circle at the centroid of the fingertip that crosses the hand pixels. The finger angle is estimated to be the same as the angle of the radial line that bisects the chord. While this method appears to be successful on all images, according to the authors, its underlying assumption is that the image contains the entire hand. When images contain only a single finger down to the PIP joint, the results are mixed. In addition, several parameters must be adjusted for any given data set. The ideal skin color vector must be adjusted for lighting conditions and test subjects. The color angle threshold depends on lighting conditions and can frequently include background regions in the finger CC. The size of the neighborhood and the percentage of the neighborhood required for a pixel to be classified as a fingertip vary depending on the shape of a person's finger and the scale of the finger with respect to the image. The final parameter, the radius of the circle used to find the finger direction, varies with scaling as well as with the location of the fingertip centroid. If any of these parameters is adjusted incorrectly for a given image (or a given test subject or lighting condition), the finger will be found in the wrong location and with the wrong orientation. Finally, once the fingers have been located, the rest of the procedure is the same as for the Scaled Rigid-Body Transform, meaning that this method has the same problems with skin deformation.

Only a few examples exist of finger image registration outside the realm of fingernail imaging for finger force detection. One, a method of biometric authentication using fingernail structures was created [34], although this technique required images of the entire hand to begin, and the nails were registered using an ad hoc method that made several assumptions about the shape and size of the nail relative to the finger. This method would likely suffer from the same deficiencies as the Hand-to-Finger Transform if applied to fingernail imaging.

Other image registration techniques have been considered and rejected for various reasons. The Lucas–Kanade iterative approach [35], for example, uses the gradient of the image to align two images. One of the key requirements of this method, however, is that the object to be registered must have approximately the same pixel intensities from one

image to the next, the differences being due to Gaussian noise. Fingernail images during force application do not exhibit this characteristic as the color patterns change significantly depending on the force magnitude and direction.

A novel registration method has been developed [36, 37] that iteratively uses Active Appearance Models (AAM) to register all of an individual's data, compensating for scaling, translation, and rotation. The technique requires a group of training images, which should be representative of the modes of variation of the data set. A set of common landmark points are identified on each of the training images. A statistical model of Shape variation is generated by aligning each set of points to a common frame of reference and applying a Principal Component Analysis (PCA) to find the modes of Shape variation. Each training image is then warped to the mean shape, and the gray-level information is extracted. These values are then normalized, and PCA is applied to find the modes of Texture variation. The Shape and Texture parameters for each training image are next combined into a single Appearance vector, and PCA is applied a third time. This final PCA generates the modes of Appearance variation. Finally, a Search Model is created by correlating the pixel intensities to the Appearance and pose parameters. Given a new image, this Search Model may be used to estimate the appropriate Shape, Texture, and Appearance parameters as well as the required position, scale, and orientation.

AAMs have been used in a wide variety of applications. Some of the earliest applications were to register images of faces [38–41] and electrical components [38]. Later applications, particularly in medical image analysis, include knee MRI [38, 42], cardiac MRI [40, 43], dental radiographs [44–46], vertebra MRI [47], eye OCT scans [48], abdominal organs [49], and red blood cells [50]. The general AAM procedure has been applied to the problems of counting people [51], performing expression recognition [52, 53], synthesizing speech [54], and synthetically aging individuals in images [55]. As far as the author is aware, no attempt has been made to register fingernail images using AAM.

1.3 Modeling and Optimization

Multiple force prediction models have been used throughout the history of fingernail imaging. Each has drawbacks and advantages. In this work, previous models have been expanded to accommodate three-dimensional force prediction. In addition, new models have been developed. All are evaluated herein. In addition, the light color is investigated further to determine whether it may be optimized. The effect of the design of the grid of calibration points is characterized.

1.3.1 Force Prediction Models

Several models relating pixel intensity to the magnitude of finger force have been implemented in the past. A generalized least squares technique, which forms a weighted linear least-squares model relating a single force direction to all of the pixel intensities in the finger, was found to have an RMS error in all directions of 0.2 N to 0.6 N [3]. This model relies on locally-weighted linear regression [56] to fit the response of each pixel to the force before determining the model coefficients.

A multivariable least squares model was used in later work [26]. In the newer model, all of the data were assumed to form a linear fit between pixel intensity and force in each direction. This model was found to have an RMS error of 0.3 N in all three directions. However, the accuracy of this fit was dependent on the registration method, in addition to overfitting the data. Registration using anything but the Scaled Rigid-Body Transform degrades the performance of this model. The registration method was not correctly aligning the fingers as pixels near the edges of the finger and around the edge of the nail were the most relevant for force prediction.

The EigenNail method [27] was used to classify the direction of force. This technique uses the eigenvectors of the pixel data set to linearly separate the classes (i.e., the different directions of force). It was shown to correctly classify forces in all directions with at least 70% accuracy, which improved to 90% upon application of a second level of classifier.

1.3.2 Lighting

Different lighting methods have been used throughout the history of fingernail imaging. In early experiments [3, 26, 27], a diffusely reflective dome with several yellow-white LEDs mounted around the outside reflected light onto the finger, attempting to create diffuse illumination, as illustrated in Fig. 1.4(a). This dome was mounted around the camera so that light emanated from approximately the same location as the camera and placed near the finger, as shown in Fig. 1.4(b). Later experiments have used arrays of LEDs in different configurations, placed above and behind the finger. Different colors of LEDs have been used, including bright white, cool white, and green. The inclusion of green LEDs is motivated by work with the fingernail sensors. The wavelength of light emitted by the LEDs has been studied [5] and green light was found to correlate with the optimal response rate of the sensors. In addition, past work has shown that the green channel of the camera correlates most effectively with the force [57].

More motivation to investigate the effect of lighting comes from the design of the camera's CCD (charge-coupled device). As shown in Fig. 1.5, the Green photodetectors (G)

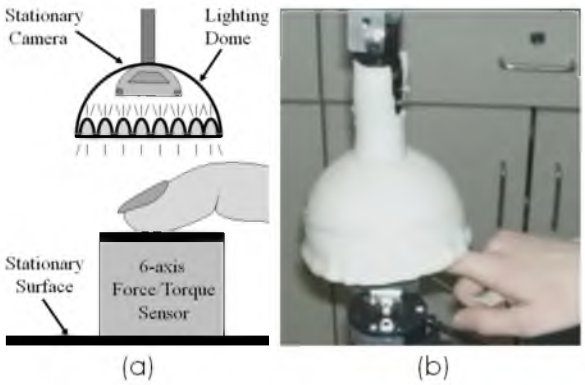

Fig. 1.4. Old lighting method. (a) A diffusely reflective dome is placed above the finger. An array of cool white LEDs is placed around the rim of the dome, shining upward, so that the light reflects off the dome and down onto the finger. (b) The lighting dome is placed close to the finger and images are recorded as the forces are exerted.

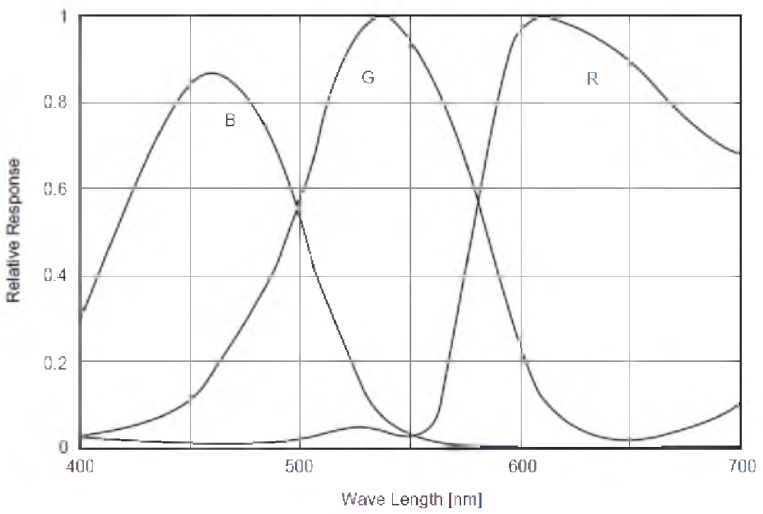

Fig. 1.5. Camera CCD photodetector response from the Sony CCD documentation [58]. The green photodetectors respond to a wide range of wavelengths, meaning they might saturate more easily under white light than under green.

respond to a wide range of wavelengths, indicating that the green channel of the RGB images includes more than just green light and may be saturating prematurely in the presence of white light. Since prior work has found that the green channel is most correlated to the force response [4], such saturation is undesired. If green LEDs were used to illuminate the finger, a narrower spectrum of light would be available to the photodetectors, and the green channel might be more sensitive to the effects seen in previous work.

1.4 Summary

The aims of this work are threefold. The first objective is to develop and characterize an automated calibration routine, including novel force control on the interaction between a robot with a flat-surface end effector and the human fingerpad. The second goal of this work is to create and investigate the effectiveness of a new registration method using Active Appearance Models for aligning fingernail images. This work's final objective is to optimize the calibration process to improve the force prediction accuracy through adjusting the lighting, designing appropriate calibration trajectories, and selecting effective force prediction models. With these tools available, it is expected that a researcher will be able to implement a grasping study that allows the measurement of interaction forces using fingernail imaging.

The following three chapters are each prepared for submission to journals appropriate for their content. Chapter 2 discusses the design, development, and characterization of a novel force controller for the automated calibration routine. This controller is specifically designed to control force during interactions between a flat-plate end effector and the human fingerpad. The modeling choices are shown and the static and dynamic calibration procedure is detailed. The controller design process is outlined, and the implementation on a Magnetic Leviation Haptic Device (MLHD) is demonstrated. The resulting controller is analyzed. The controller is applied to a group of test subjects for the purposes of imaging calibration, and the force prediction results are shown.

Chapter 3 details the new registration method. Using Active Appearance Models is a novel approach to fingernail image registration. This technique includes the design and characterization of relevant contours to retain important shape and texture information for registration. In addition, an optimal landmark point density is determined, and the effects of color-processing regime are characterized. The effect of these modifications on registration and force prediction accuracy are investigated.

In Chapter 4, the optimization of force prediction modeling, lighting choice, and calibration grid are outlined. The effect of each of these modifications on registration accuracy (where applicable) and force prediction error are analyzed.

Various materials related to each of the three papers are provided in the appendices. Appendix A contains more specific information about the use of the MLHD API (application programming interface) in the controller as well as the system identification procedures used to determine the static and dynamic properties of the flotor and camera system. The model parameters are concretely defined, including more thorough diagrams than could be

included in the journal article format. In addition, observability and identifiability of the identification data sets are addressed. Further information regarding the performance of the controller is also presented, including the gain margins, phase margins, and bandwidth for each of the six force/angle directions.

Appendix B explains, in further detail, the smoothing procedure used on the hand-selected nail contours. Minor details, such as specific choices relating to the formation of the AAM, are included in this section. This appendix also presents further material regarding the behavior of the AAM, including analyses of the covariance of Shape, Texture, and Appearance, the modes of variation for each subject, and so forth.

In Appendix C, additional details regarding the calibration grids are included. The layout of the two types of grids is given as well as the method used to divide the grids into regions for calibration.

CHAPTER 2

3-D FINGERTIP TOUCH FORCE PREDICTION USING FINGERNAIL IMAGING WITH AUTOMATED CALIBRATION

2.1 Abstract

This paper presents an automated routine for calibrating a fingernail imaging system with the intent of predicting fingerpad forces. The system uses a Magnetic Levitation Haptic Device to apply forces to the human fingerpad while recording images of the nail and surrounding skin. A novel force controller is implemented to interact stably with the human fingerpad. The data are used to calibrate a principal component regression model relating pixel intensity to three-dimensional force. Using data from this automated routine, this model simultaneously predicts three-dimensional force with an RMS error of 0.56 ± 0.03 N.

2.2 Introduction

Force estimation using fingernail coloration was introduced by Mascaro and Asada [1]. This concept relies on the coloration effect achieved by the movement of blood in the tissue beneath the fingernail. The interaction of the fingerpad with other surfaces causes blood to evacuate from some areas of the finger while pooling in others. Since the fingernail is essentially transparent, this coloration change can be seen clearly in images of the fingernail and surrounding skin. In this paper, an automated calibration method for fingernail imaging is presented that results in full three-dimensional force prediction. Prior calibration methods required a test subject to exert the forces and only estimated the force in one direction at a time. Thus, the new calibration procedure presented here results in a more robust model that is able to more accurately predict arbitrary shear and normal forces on the human finger.

2.2.1 Background

The first fingernail sensor [59] was designed to be mounted on the test subject's fingernail, in much the same way as an artificial nail. It consists of an array of LEDs and photodetectors that illuminate the fingernail and measure the coloration effect. For calibration, the test subject was guided to apply a series of desired forces on a force sensor while the fingernail sensor simultaneously recorded data. This calibration data were used in the creation of a model that could be applied to future measurements to predict forces. The original calibration was able to predict force magnitude and direction over a range of shear force up to 2 N with an error of 0.5 N and normal force up to 3 N with an error of 1 N. The fingernail coloration effect was found to have patterns common across all people [2]. This commonality justifies using this method to estimate finger force.

Two major problems with the on-nail sensors are the manufacturing cost and the resolution of the sensors. Due to variability between fingers, the sensors must be individually manufactured to fit each test subject. Even when the sensors are fit perfectly, there is limited space for photodetectors and LEDs. Additionally, the fingernail saturates near 6 N, while the surrounding skin can transduce forces up to 10 N. A new method was proposed using a high-resolution digital camera [57]. The data set available from such an approach is much larger. In addition, the method may be applied to any finger, regardless of size or shape. This method also allows the surrounding skin to be imaged and used in the force estimation. The calibration process was much the same as before, with the test subject asked to exert forces to cover the force space. Using the digital camera, the calibration error in normal force was reduced to 0.3 N over a range up to 10 N. In other work, force was estimated in one direction at a time with accuracy of 0.1 N in all directions over the ranges of normal force up to 10 N and shear up to 2 N [3]. Discrete finger force directions were estimated with an accuracy of 90 percent without any individual calibration [27]. The prediction accuracy held even as the resolution of the images was reduced to 10×10 pixels. This increase in accuracy comes with a price, however. The lighting must be consistent for the imaging method to function correctly. Obviously, the fingernail must remain in view of the camera. While calibration may currently be performed in a controlled setting, it will be necessary to find solutions to these problems before the imaging method can be used in other environments.

An automated calibration process requires a controller capable of applying forces in three dimensions to the human fingerpad. While many techniques exist for controlling force, research into the application of controlled force to the human fingerpad has been

limited. Finger dynamics have been studied using single-axis actuators [21–23,60], although all of these used position control and measured the output forces. A high-performance three-axis force controller was developed for a haptic device to apply three-dimensional forces to the fingertip [24]. However, this method placed the fingertip in a thimble, which applied the forces to the finger as a whole rather than to the fingerpad. Thus, no distinct shear forces were being applied. One true three-dimensional force controller applying forces between the human fingerpad and a flat surface was implemented using two connected Novint Falcons [25]. The controller had a settling time of 0.25 s and was able to track desired trajectories with RMS errors of 0.05 N or less in all three force directions. While the RMS error would be sufficient for fingernail imaging calibration, a faster settling time is desired for calibration of fingernail imaging. In addition, no maximum force range was mentioned in the paper. The dynamic behavior of the fingerpad and the coupling of forces in the three axes due to finger geometry and frictional constraints make controlling these forces a unique problem.

2.2.2 Current Work

In prior experiments, calibration was performed using a GUI that guided the test subjects through the desired force space and asked them to provide the expected forces. Several limitations of this method have led to the development of this new automated calibration routine. First, nearly all subjects required time to become familiar with the calibration procedures, slowing the entire process. Second, subjects had some difficulty controlling two directions of force at a time. It was found that independently controlling normal force and shear force was nearly impossible. It is logical to assume that adding more variables, such as shear torque, contact angles, and finger joint angles, would likely complicate the process beyond the ability of most subjects. Third, after a few minutes, it became increasingly difficult to maintain the level of force on the finger, especially at large shear and normal force combinations. It is desired to reduce or eliminate this fatigue on the test subjects. Fourth, the human subjects required time to process the change in desired force and to exert that force, which prolonged the calibration and added to the fatigue. An automated calibration system simplifies the process and answers all of these difficulties. Familiarization time with the calibration procedure is reduced. Control of the variables is performed by computer. The calibration can be made mostly passive so that the subjects are more relaxed throughout the testing. Since the calibration system is in control, changes in force may be performed much more quickly, and desired force levels may be held more

reliably. This allows thousands of images to be collected where previous calibration methods required multiple sessions to record little more than 300 images.

2.3 Calibration Procedures

2.3.1 Experimental Apparatus

Fig. 2.1 illustrates the experimental apparatus. To perform the automated calibration, a Magnetic Levitation Haptic Device (MLHD) [61, 62] is used. The MLHD consists of a flotor supported by the magnetic field generated in six Lorentz coils. The flotor can translate freely in a 12 mm-radius sphere while rotating up to 8° in any direction. It is capable of exerting up to 20 N of three-dimensional force and 4 N · m of three-dimensional torque. While these ranges are adequate to apply force to the human fingerpad and perform the desired calibration, they are insufficient to cause damage to the finger, and so the experiments will not endanger the test subjects. The MLHD has the advantages of high fidelity (< 2 µm resolution) and high bandwidth (130 Hz) [63]. These properties allow for excellent static and dynamic characterization of the force/coloration relationship.

An ATI Nano17 6-axis Force/Torque sensor is attached to the flotor of the MLHD to measure the contact force. A rubber surface is attached to the sensor to provide comfort, and a 3 cm square of two-sided tape is affixed to improve frictional behavior. The sensor is capable of detecting normal forces up to 17 N and shear forces up to 12 N with a resolution of 0.78 mN. This range and resolution is sufficient for the calibration as the intended ranges are 10 N normal and 5 N shear.

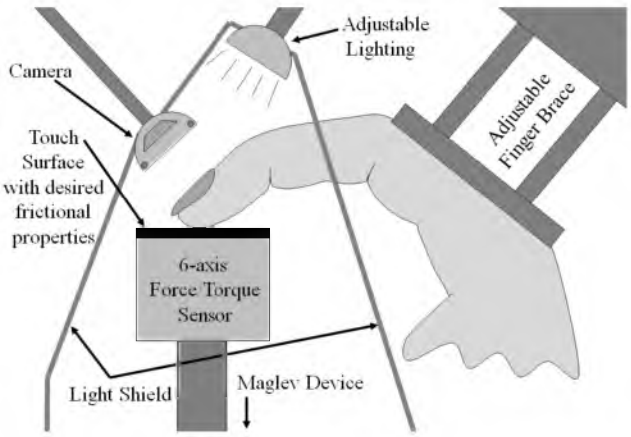

Fig. 2.1. Schematic of calibration setup. The flotor exerts force against the fingerpad with a flat contact surface, while the finger rests against the restraint to guide the approximate angles of the joints. The camera is positioned above the nail with the light source above and behind the finger. The light shield blocks glare from ambient light.

A Point Grey Research FLEA video camera is mounted over the finger to record images during the calibration. This camera can record 1024 × 768-pixel RGB images at 30 Hz. The camera is attached to a static frame since the location of the finger does not vary substantially during calibration. The camera uses a Tamron A201118 lens whose aperture and focal length can be manually adjusted to find the optimal light level and focus for each set of calibration images.

A lighting box is placed above and behind the finger to provide uniform illumination during calibration. This box consists of an array of 140 LEDs with a piece of tracing paper (3-lb. weight) to diffuse the light and reduce the shadowing effect due to the internal structure of the LEDs. The box is placed on a gooseneck so that it may be adjusted for individual variation prior to each experiment. Positioning the light source above and behind the finger ideally eliminates any glare on the nail, allowing only that light which penetrates into the nail bed to reflect back to the camera. Two lighting boxes are created, one containing white LEDs and the other containing green. It is eventually desired to investigate the effects of lighting color on the force prediction results as other research has shown that on-nail sensors react best under green light [64]. White light is chosen as a reference color since all fingernail imaging up to this point has been performed under white light. A light shield is mounted over the camera, the finger and the MLHD to block ambient light.

Past research [59] has found that the metacarpophalangeal (MCP) joint angle has no effect on the coloration of the finger. The distal interphalangeal (DIP) and proximal interphalangeal (PIP) joints, on the other hand, do affect the blood flow to the finger (and hence the coloration). Direct control of either the DIP or PIP joints without affecting blood flow or obscuring the camera's view of the finger has thus far proved impossible. However, they can be controlled in the sense that the angle of the proximal phalanx can be controlled relative to the force sensor. If this angle is set correctly, the finger is positioned to encourage test subjects to maintain a constant angle in each of the DIP and PIP joints. To position the finger during calibration, a restraint device is implemented using a Roylan Static Progressive Finger Flexion Splint. The straps, which would ordinarly be used to hold the splint on the finger, restrict blood flow and so are removed. Rather than restrain the finger's movement, the splint is used to guide the subject in finger placement.

Although any of the fingers may be calibrated using the standard calibration position, doing so with the thumb would be awkward since the subject would be required to stand for an extended period of time. This is because of the shape of the hand and the location

of the thumb relative to the rest of the fingers. Instead, an alternate calibration position is designed, at 90° to the standard position (Fig. 2.2). A separate restraint is attached to the frame that guides the thumb into place, and a separate end effector is designed with a vertical pad to contact the thumb. The data presented in this paper corrects for this rotation so that F_z always corresponds to normal force on the finger, F_x corresponds to lateral force, and F_y corresponds to longitudinal force. In this way, the forces on the thumb and the finger may be directly compared, in spite of the configuration differences.

A picture of the calibration setup is shown in Fig. 2.3(a). The finger is placed in a

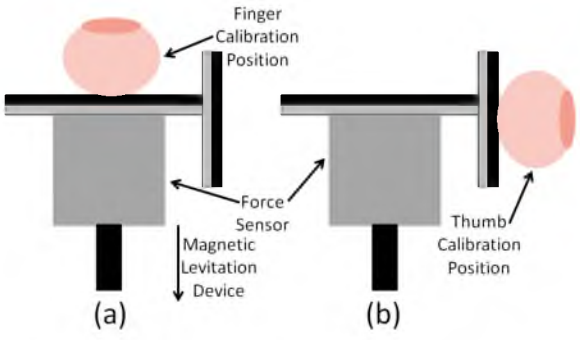

Fig. 2.2. Schematic of finger positions. (a) The index finger is calibrated so that the x-, y- and z-axes are aligned with the flotor's axes. (b) To accommodate the shape and position of the thumb on the hand, it is calibrated in a position at 90° relative to the other fingers. The forces are rotated so that F_z always corresponds to normal force.

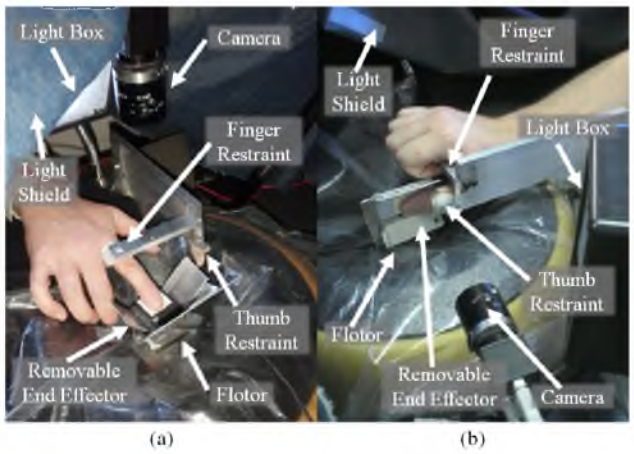

Fig. 2.3. Photos of calibration equipment. (a) Calibrating the index finger. The camera is positioned above the finger, with the light box above and behind the finger. The restraint positions the finger with respect to the flotor, which contacts the fingerpad. The light shield has been partially removed to allow this picture to be recorded. (b) Calibrating the thumb. While the thumb is oriented at 90° to the finger calibration position, the relative positions of all other components are similar.

restraint that controls the proximal phalanx angle and positions it above the MLHD. The camera is stationed above the finger, while the light source is positioned above and behind the finger. The thumb calibration method is shown in Fig. 2.3(b). The thumb is calibrated at 90° to the index finger, with a separate restraint and a separate end effector. The camera and light sources are placed in different locations, maintaining their locations relative to the nail.

2.3.2 Interface Design

The finger restraint merely acts as a guide for the proximal phalanx. Due to the motion afforded by the finger joints, the test subject must still provide some active input to the system. The MCP joint has a range-of-motion (in abduction/adduction) of 35°, and the average distance from the MCP to the end of the finger is 8 cm [65]. If the MCP is held stationary but free to rotate in this direction and the contact location is assumed to be at the end of the finger, the flotor would need to translate approximately 11.4 cm to remain in contact, far beyond the workspace limits.

The subject is therefore provided with a graphical interface as a guide to maintain the flotor within the workspace. A short period of instruction is given before each experimental session to familiarize the subject with the GUI (Fig. 2.4), which consists of an L-shaped crosshairs marking the flotor's current (x, y) coordinates and a circle, centered at the origin, representing the boundary of the workspace at the current z-coordinate. In addition, a white L-shaped crosshairs marks the origin of the workspace, and a white circle represents the maximum radius of the workspace. As the flotor's z-coordinate increases from 0 mm to

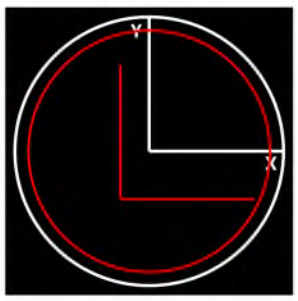

Fig. 2.4. Graphical user interface for automated calibration. The white xy axes indicate the origin of the flotor's workspace, while the red axes indicate the flotor's current location. The red circle indicates the radius of the workspace at the current z-coordinate, while the white circle indicates the maximum radius, at $z = 0$. The circle and axes are red when $z > 1$ mm, magenta when $z < -1$ mm, and green when $|z| \leq 1$ mm. When this image was generated, the flotor was near $(-3, -4, +4)$ mm.

+12 mm, the circle's diameter decreases, and as the flotor's z-coordinate decreases from 0 mm to −12 mm, the circle's diameter also decreases, as if the subject is viewing the horizontal cross-section of the spherical workspace. To aid subjects with understanding the z-coordinate, the circle changes colors, green when $|z| \leq 1$ mm, red when $z > 1$ mm, and magenta when $z < -1$ mm. In this way, the user receives visual cues indicating how to adjust position in all three axes simultaneously, to assist in the task of maintaining the flotor's position near the center of the workspace.

2.3.3 Experiment Design

To test the force controller, data from 17 subjects is collected. The test subjects include 13 males and four females. Twelve subjects are Caucasian, two are of east Asian descent, and three are of Middle-Eastern origin. Each subject sits for a period of two hours while images and forces are recorded on both the index finger and the thumb. The images and the corresponding forces are recorded only once at each force level (a frequency of approximately 2 Hz), which is used to calibrate the force prediction model. At the same time, information such as the position and orientation of the flotor, the force input to the flotor, and the interaction force between the flotor and the subject's finger, are all recorded at a frequency of 1 kHz. This much larger data set is then used to investigate the operation of the force controller.

The test sessions proceed as follows. Four data sets (finger/LED color combinations) are collected from each subject: (1) Index/White, (2) Thumb/White, (3) Index/Green, and (4) Thumb/Green. Each subject is introduced to the task and the visual cuing system. The desired force space is divided into nine sections, and the force controller traverses each of these sections three times for each subject/finger/LED combination. (The overall order is randomized to minimize the effects of learning and fatigue on a particular section.) This results in 108 data files per subject ($9 \times 3 \times 2 \times 2$), each containing between 6000 and 92 000 data points. When stored as ASCII .txt files, this requires nearly 17 GiB of disk space.

In addition, two different types of calibration grid are used. The points may be arranged in either an (x, y, z) Cartesian grid or an (r, θ, z) cylindrical grid. The type of grid is assigned randomly to each subject/finger/LED color combination. Each grid contains approximately the same number of points, although they cover the space in a different fashion.

2.3.4 Image Registration

Active Appearance Models (AAM) [39] are used to register the images. Each finger's images are used to form an AAM following a method developed previously [36, 37]. The

Search Model is used to register all images pertaining to that particular finger. Once registered, the images may be used to form a force prediction model.

2.4 Control Design

2.4.1 Controller Requirements

The primary requirement of the force controller is that it be accurate. As the range of normal forces on the fingers (other than the thumb) in a typical five-fingered grasping experiment [15] is less than 10 N, it may be assumed that two-fingered grasping would involve forces up to 10 N on each finger. Therefore, it is desired to apply normal forces up to 10 N and shear forces up to 5 N with a resolution of 0.1 N. It is therefore desired that the steady-state error of the force in any direction be less than 0.05 N.

A secondary requirement is that the calibration process be as fast as possible. The test subject is not required to control the force, as in previous calibration processes, which should reduce the effects of fatigue. A new task has been added, in that the need to actively resist the forces, in order to keep the MLHD end effector within the workspace, is fatiguing over an extended session. The larger number of desired forces, combined with the larger magnitude of total force, is expected to create nearly as much fatigue as the manual calibration processes used in prior experiments. Therefore, the calibration should proceed as swiftly as possible. However, the speed of the process is limited by the hemodynamics of the finger, which have time constants of between 0.1 s to 0.4 s [66]. Therefore, the force controller should have a two percent settling time of no more than 0.1 s so that calibration time can be minimized.

Safety is obviously a major concern. As mentioned previously, the MLHD is incapable of injuring the test subjects through impact loading. With IRB oversight, fatigue-related injuries are carefully controlled through experimental design and in-experiment monitoring. Damage to the MLHD itself is prevented during normal operation through software limits. The most relevant problem during the experiments is that the controller becomes unstable and the flotor moves outside its workspace. In this case, the flotor needs to be raised again, the controller restarted, and the test reinitialized from the appropriate point. Thus, falling out of the workspace is an inconvenient delay rather than a danger to the test subject or the equipment.

The MLHD API provides commands that allow the user to specify any of the following control inputs: (1) electrical current in each of the six Lorentz coils, (2) position and orientation of the flotor, (3) translational and angular velocity of the flotor, or (4) the forces and torques exerted on the flotor by the Lorentz coils. A force controller was

developed using the position commands as an inner loop [67]. However, the authors have been unable to eliminate some high-frequency dynamics present in that controller, leading to the supposition that they are due to some unknown dynamics of the internal position controller. The controller could be developed using the current commands, requiring detailed identification of the current-to-force system properties. While it would be possible to develop a force controller using the velocity commands, this approach is unintuitive and needlessly complex. Since the force commands provide simple open-loop force control, they can be used in concert with the force sensor to create a closed-loop controller that applies forces to the fingerpad.

A nonlinear feedback controller with an underlying PID controller is used in this work. To design this controller, it is necessary to develop a model of the open-loop response of the system.

2.4.2 System Model

The flotor is modeled as a 6-DOF rigid-body lumped-parameter mechanism, able to freely translate and rotate in x, y, and z (Fig. 2.5). The angles reported by the MLHD are assumed to be ZYX Euler angles, that is, that the rotation matrix of the flotor relative to

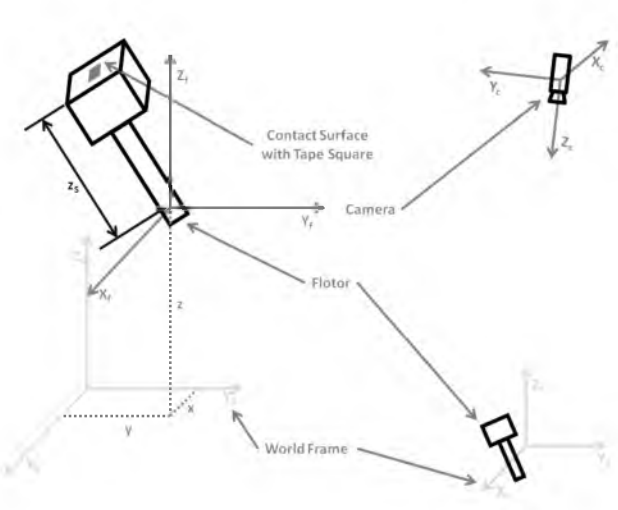

Fig. 2.5. Schematic of kinematic calibration. On the left is a close-up view of the flotor, showing the World Frame (X_0, Y_0, Z_0) in light gray, the flotor coordinates (x, y, z) and unrotated Flotor Frame (X_f, Y_f, Z_f) in dark gray, and the rotated flotor in black. The distance from the Flotor Frame origin to the contact surface z_s is indicated. On the right is shown the flotor and camera (both in black) relative to the World Frame (again in light gray) and the Camera Frame (X_c, Y_c, Z_c) in dark gray. During kinematic calibration, a small piece of tape provides fiducial marks on the contact surface while the flotor moves through the workspace and the camera records images.

the base frame is represented by $R_z(\theta_z)\, R_y(\theta_y)\, R_x(\theta_x)$. Since these angles are all small (less than 8°), however, the order of rotation is assumed to be irrelevant. The flotor has a mass m_f and a center of mass (defined in the frame of the flotor) located at $\mathbf{r}_f = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$. The inertia matrix of the flotor is defined as $\mathbf{I}_f$, likewise with regard to the frame of the flotor.

When performing parameter identification, it is assumed that the finger is not in contact with the flotor. Then, the forces $\mathbf{F}_m$ and torques $\boldsymbol{\tau}_m$ on the flotor may be written in the following form:

$$\mathbf{f}_n = \mathbf{I}_n \mathbf{a}_n + \mathbf{v}_n \times \mathbf{I}_n \mathbf{v}_n \tag{2.1}$$

$$\mathbf{f}_n = \begin{bmatrix} \boldsymbol{\tau}_m \\ \mathbf{F}_m \end{bmatrix} \tag{2.2}$$

where $\mathbf{I}_n$, $\mathbf{v}_n$, and $\mathbf{a}_n$ are the spatial inertia, spatial velocity, and spatial acceleration of the flotor, respectively. (See Section 14.3 in [68] for more information.) This equation may be rewritten in a linear least squares form:

$$\mathbf{f}_n = \mathbf{A}_n \boldsymbol{\phi}_n \tag{2.3}$$

where $\mathbf{A}_n$ contains terms relating to the position, velocity, and acceleration, and $\boldsymbol{\phi}_n$ is a vector containing the unknown parameters (m_f and the components of $\mathbf{r}_f$ and $\mathbf{I}_f$).

2.4.3 System Identification

To fully characterize the system, the parameters of the model must be defined. The location of the base frame is determined. Then, the wrench formula (2.3) may be used to determine the flotor inertial parameters. Finally, the finger force may be added to the system model to determine its effects. Only then can an appropriate controller be designed to interact with the finger and compensate for the nonlinearities present in the system.

2.4.3.1 Static Parameter Identification

First, the coordinate systems must be established. The world frame is defined such that, when the flotor is at the zero position, the world and flotor frames are aligned. The flotor frame is defined such that the axes are parallel to the force sensor's axes, with the origin located at the center of rotation in all three axes when the flotor is at $(0, 0, 0)$. The kinematic calibration is used mainly to estimate the location of the flotor frame origin relative to the contact platform.

The calibration procedure is illustrated in Fig. 2.5. A piece of white two-sided tape is affixed to the contact platform. The previously-calibrated camera is used to record an

image at each of 120 poses throughout the workspace. The flotor pose is measured by three planar position sensitive photodiodes paired with LEDs mounted on the flotor and reported through the MLHD API function. The four corners of the tape are used as the reference points in the images. Canny edge detection [30], in concert with the standard Hough Transform, is used to identify the corners. In two of the images, one side of the tape is out of the frame, meaning that 476 data points are available. Given that each data point is represented by the camera's (u, v) coordinates, this gives two equations per data point, for a total of 952 equations in the 15 unknowns.

The parameters that are to be identified during this process are (1) the six components of camera pose (location and orientation) relative to the base frame, (2) z_s, the vertical coordinate of the contact platform within the flotor frame (i.e., the distance between the origin of flotor actuation and the fingerpad), and (3) the x- and y-coordinates of the four tape corners within the flotor frame. The initial parameter values are estimated through manual measurements. The task variables (i.e., the camera coordinates u and v) are scaled using their respective standard deviations, while parameter scaling is accomplished using the Euclidean norm of each column of the calibration matrix [69]. The Gauss-Newton method converges to a residual of less than 0.01 after 6 iterations. The fingerpad contact surface is found to lie $z_s = 48.1 \pm 0.5\,\text{mm}$ above the origin of flotor actuation. The other parameters fit well with the measured values. The scaled calibration matrix has a condition number of 53, within the range suggested by Schröer [70], indicating that the parameters are identifiable from this calibration set.

2.4.3.2 Inertial Parameter Identification

Second, the flotor's inertial parameters must be identified. A series of wrench commands that should excite all of the model parameters is sent to the MLHD and the position, orientation, and input wrench are recorded for n time steps during these commands. Then, the n data points are substituted into (2.3) to produce a set of $6n$ equations in 10 unknowns (the inertial parameters). Finally, this set of equations is solved to calculate the inertial parameters. Given this overview, the details are now shown:

For data collection, a series of force/torque commands is sent to the MLHD designed to excite all of the modes of movement so that the parameters may be identified. The flotor pose is measured using the MLHD API, with numerical differentiation used to calculate velocities and accelerations. A series of step inputs tracing out a grid around the workspace is selected that covers the range $-6\,\text{mm} \leq k \leq 6\,\text{mm}$ and $-4° \leq \theta_k \leq 4°$ at intervals of 3 mm and 2° (where $k \in \{x, y, z\}$). The wrench applied by the MLHD to the flotor, as well

as the flotor's position and orientation, are recorded at each time step. A short section of the inputs and outputs is shown in Fig. 2.6. A total of 80 s of data are collected as the MLHD moved through this trajectory at a rate of 1 kHz.

Since (2.3) is linear in the parameters, ordinary least squares may be used to solve for the parameters. These six equations (with the 80 000 data points gathered) provide 480 000 equations in the 10 unknown parameters. These equations are solved using Matlab's matrix algebra solver. The values found using this method are reported in Table 2.1. Given that these numbers are approximately the same as those reported during the design of the initial MLHD prototype [61], the authors are confident in the results reported here.

2.4.4 Controller Design

With a complete model, the controller may be designed to appropriately compensate the system. The controller is a standard PIVF force control scheme [71] with nonlinear feedback compensation, as shown in Fig. 2.7. Defining the force error as $\Delta\mathbf{F} = \mathbf{F}_d - \mathbf{F}$, the

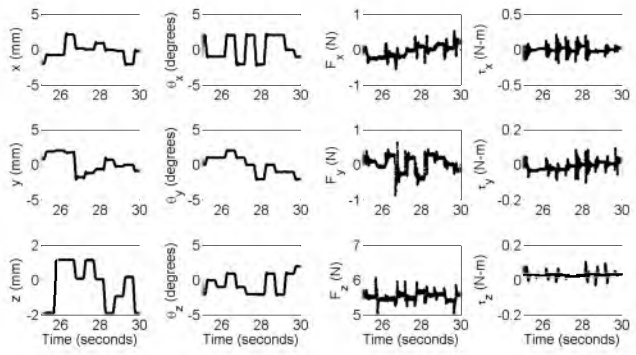

Fig. 2.6. Selection of flotor dynamic parameter identification data. Time is given on the x-axis in all plots. In the first column, the flotor position coordinates x, y, and z are shown. The second column contains the flotor angles θ_x, θ_y, and θ_z. The forces exerted by the Lorentz coils on the flotor F_{xm}, F_{ym}, and F_{zm} are given in the third column. In the fourth column are plotted the torques exerted by the Lorentz coils on the flotor τ_{xm}, τ_{ym}, and τ_{zm}.

Table 2.1. Flotor Inertial Parameters

Qty	Value	Qty	Value
m_f	(0.56 ± 0.01) kg	r_y	(1.84 ± 0.01) mm
r_x	(1.98 ± 0.02) mm	r_z	(-18.7 ± 0.1) mm
I_{xx}	(0.161 ± 0.050) g $\cdot$ m^2	I_{xy}	(0.003 ± 0.004) g $\cdot$ m^2
I_{xz}	(0.046 ± 0.005) g $\cdot$ m^2	I_{yy}	(0.900 ± 0.130) g $\cdot$ m^2
I_{yz}	(0.037 ± 0.013) g $\cdot$ m^2	I_{zz}	(2.65 ± 0.36) g $\cdot$ m^2

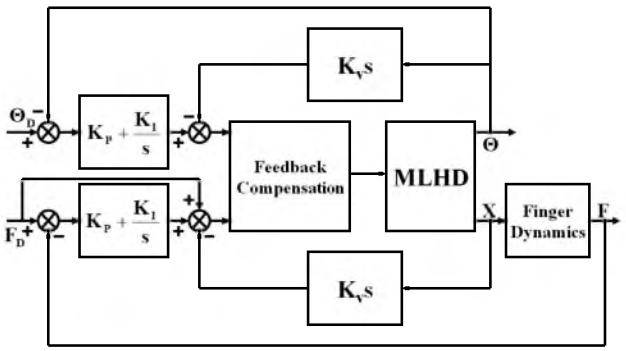

Fig. 2.7. Force controller design. The forces (in x, y, and z) are controlled in a separate loop from the angles. The two controllers join for nonlinear compensation. Since the goal is to maintain planar contact with the finger, when the controller is implemented, the desired angles $\boldsymbol{\Theta}_D$ are identically zero.

force control input $\mathbf{F}_{\text{in}}$ and rotational control input $\boldsymbol{\theta}_{\text{in}}$ may be written as

$$\mathbf{F}_{\text{in}} = K_P \Delta\mathbf{F} + K_I \int \Delta\mathbf{F}\, dt - K_V \mathbf{v} + \mathbf{F}_d \tag{2.4}$$

$$\boldsymbol{\theta}_{\text{in}} = -K_P \boldsymbol{\theta} - K_I \int \boldsymbol{\theta}\, dt - K_V \boldsymbol{\omega} \tag{2.5}$$

where $\boldsymbol{\theta}_d = 0$, since the controller will only be used to stabilize the angles. Finally, the nonlinear compensator has the general form

$$\begin{bmatrix} \boldsymbol{\tau}_m \\ \mathbf{F}_m \end{bmatrix} = \mathbf{I}_n \mathbf{u} + \mathbf{v}_n \times \mathbf{I}_n \mathbf{v}_n + \begin{bmatrix} z_s {}^0\mathbf{R}_f \hat{\mathbf{k}} \times {}^0\mathbf{R}_f \mathbf{F}_{\text{in}} \\ \mathbf{0} \end{bmatrix} \tag{2.6}$$

$$\mathbf{u} = \begin{bmatrix} \boldsymbol{\theta}_{\text{in}} \\ {}^0\mathbf{R}_f \mathbf{F}_{\text{in}} \end{bmatrix} \tag{2.7}$$

where the final term on the right-hand side of 2.6 compensates for the displacement between the flotor origin and the sensor origin. When the system parameters have been estimated exactly, the nonlinearities from the matrix $\mathbf{I}_n$ and the cross product all cancel exactly with their counterparts in (2.1), resulting in a set of dynamic equations of the form

$$\begin{bmatrix} \boldsymbol{\alpha} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{\text{in}} \\ \mathbf{F}_{\text{in}} \end{bmatrix} \tag{2.8}$$

or, essentially, six pure inertias with PID controllers in place. Small deviations in parameter estimation are compensated by the controller.

The block labeled "Finger Dynamics" represents the dynamic interaction between the flotor and the fingerpad. Although the dynamics of the fingerpad are highly nonlinear [21, 22, 60, 72], for the purposes of simulation and initial gain estimation, a linear approximation is used. It is assumed that the finger can be represented by a simple stiffness-damping

lumped-parameter model of the form $\mathbf{F}_s = b_f\mathbf{v} + k_f\mathbf{p}$. For the initial controller design and simulation, estimates of $b_f = 5\,\mathrm{N}\cdot\mathrm{s}\cdot\mathrm{m}^{-1}$ and $k_f = 2000\,\mathrm{N}\cdot\mathrm{m}^{-1}$ are used. These rough estimates are calculated using very simple force-displacement-velocity measurements. Due to the complex nonlinear behavior of the finger, including stiffness and damping in the skin and the joints, this linear model is used as a first approximation to find initial estimates of the controller constants. These controller constants are then adjusted once the controller is implemented, as described in Section 2.5.

Prior to implementation, the controller is simulated to verify desired operation and find initial estimates for the controller constants. The results of step inputs in each direction are shown in Fig. 2.8. Note that in this figure, the steps occur at different times, but the plots have been aligned to allow for ease of comparison and to reduce the number of separate plots that must be created. Each force direction settles within 0.05 s with less than 35 % overshoot. (Note that, during fingernail imaging calibration, the subjects will experience ramp-and-hold rather than step trajectories, so the overshoot will not be as dramatic as what is shown here.) The simulation was also used to verify that the controller inputs to the MLHD do not exceed the maximum allowable values.

2.5 Results

The controller is implemented on the MLHD using C++. The initial estimates of the controller constants found during simulation are adjusted once the controller is implemented until stability is achieved. This period of gain tuning required approximately 2 hours once the controller was brought online, but this single set of gains was found to be stable for all

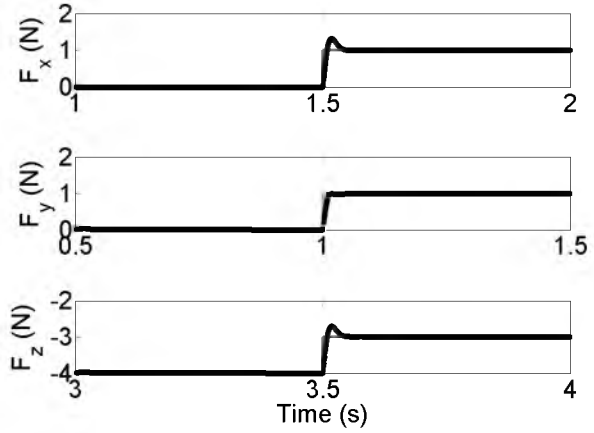

Fig. 2.8. Force controller simulation results. Note that the steps in each force direction occur at different times, but are aligned in this figure for comparison's sake. All force directions settle within 0.05 s.

19 subjects. The final set of gains used for all experiments is given in Table 2.2.

2.5.1 Controller Implementation

Several tests are performed to verify that the controller is working as designed. First, step inputs in each of the three principal force directions are applied. The response in the x-direction is shown in Fig. 2.9. The force response is found to have a 2 % settling time of 0.2 s. The overshoot is less than 7 %. The other directions show good disturbance rejection, never deviating by more than 0.4 N from the target values.

Fig. 2.10 shows the response to a step input in the y-direction. There is no overshoot in the y-direction, but the 2 % settling time is 0.25 s, which is longer than desired. The x-direction response shows a slight deviation from the desired force level, but never more than 0.5 N and settling back to the desired force within 0.05 s. The z-direction force shows

Table 2.2. Controller Gains

Dir	Gain	Value	Dir	Gain	Value
F_x	K_p	0.5	θ_x	K_p	30
	K_d	5		K_d	0.15
	K_i	20		K_i	20
F_y	K_p	0.5	θ_y	K_p	30
	K_d	5		K_d	0.15
	K_i	20		K_i	20
F_z	K_p	0.5	θ_z	K_p	10
	K_d	20		K_d	0.15
	K_i	40		K_i	0

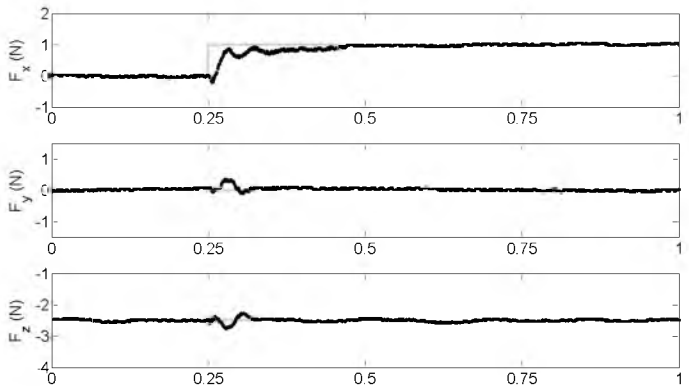

Fig. 2.9. F_x step response. There is less than 7 % overshoot, with a 2 % settling time of 0.2 s. The other two force directions deviate by less than 0.4 N from their target values.

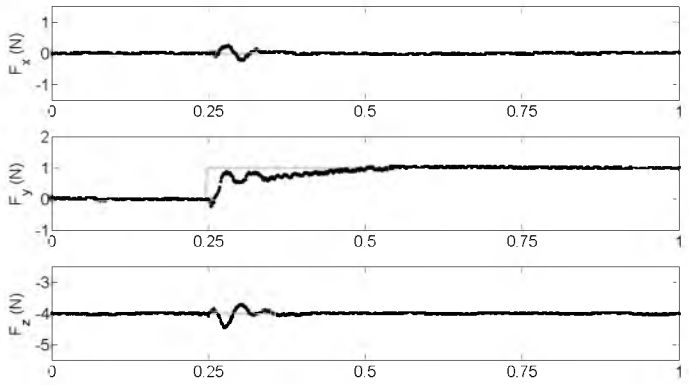

Fig. 2.10. F_y step response. The 2 % settling time is 0.25 s, somewhat longer than the goal of 0.1 s, but with no overshoot. The other two force directions never deviate from their targets by more than 0.5 N and settle within 0.05 s.

a small amount of coupling with the y-direction, oscillating between ± 0.1 N but settling to the desired value within 0.05 s.

Fig. 2.11 shows the response to a step input in the z-direction. The response has an overshoot of 25 % and a 2 % settling time of 0.09 s. The other directions exhibit good disturbance rejection, with a maximum error of 0.2 N. Both settle back to the desired force within 0.05 s.

An arbitrary step input in shear force is shown in Fig. 2.12. The response can again be seen to have a 2 % settling time of 0.25 s. Both directions show overshoot of approximately 30 %. The z-direction exhibits excellent disturbance rejection, never deviating by more than 0.4 N and settling within 0.05 s.

These results are then extended and a step is commanded in all three directions at

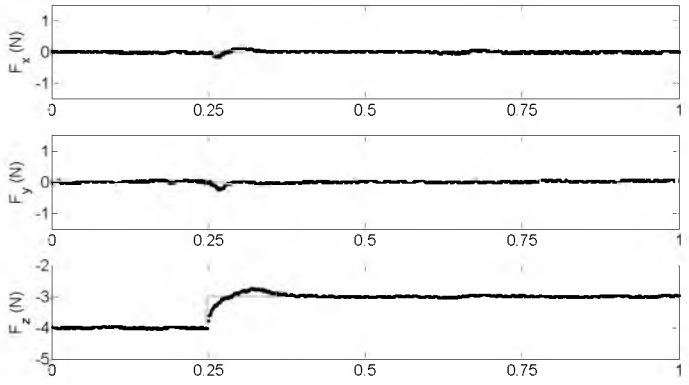

Fig. 2.11. F_z step response. The step response has 30 % overshoot with a 2 % settling time of 0.09 s. The other two directions show excellent disturbance rejection, with a maximum error of 0.2 N, and settling back to the desired force within 0.05 s.

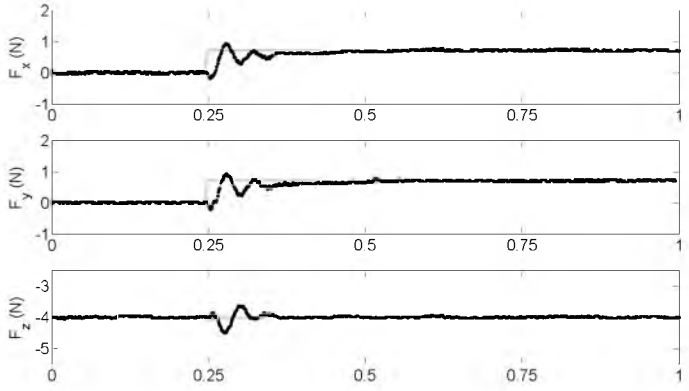

Fig. 2.12. Shear force step response. A step in both F_x and F_y is applied. Both directions settle within 0.25 s with an overshoot of 30 %. The z-direction settles within 0.05 s with an error less than 0.4 N.

once. As seen in Fig. 2.13, the applied force settles within 0.2 s, with an overshoot in the z-direction of 40 % and smaller overshoots in the other two. It should be noted that, under intended operation, the controller will never have to deal with a step input as the force will gradually transition from one level to the next.

The trajectory-tracking ability of the controller is investigated next. In this test, a constant normal force is held while a rotating shear force vector is applied to the finger. The vector rotates with a frequency of $1\,\frac{\text{rad}}{\text{s}}$. In Fig. 2.14, the controller can be seen to follow this desired trajectory with an error of less than 0.05 N and 0.1 rad. The normal force is held at the desired level with an error less than 0.05 N.

Although large variations in the finger dynamics could cause instability, the worst effect of an unstable system will be the need to realign the flotor and restart the calibration (a procedure which requires less than 1 minute). The stability of the calibration system is also

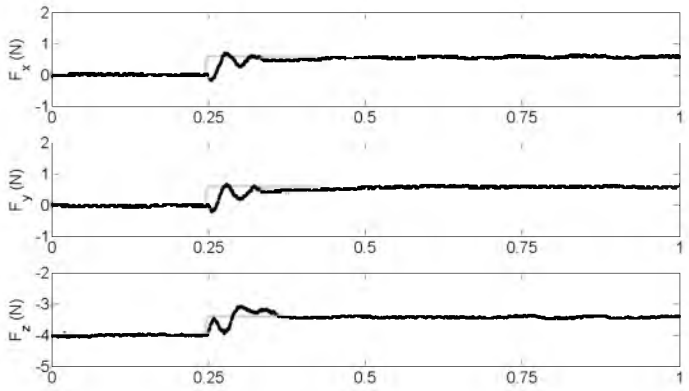

Fig. 2.13. Simultaneous step response in all three directions. The forces have an overshoot less than 40 % while settling within 0.2 s.

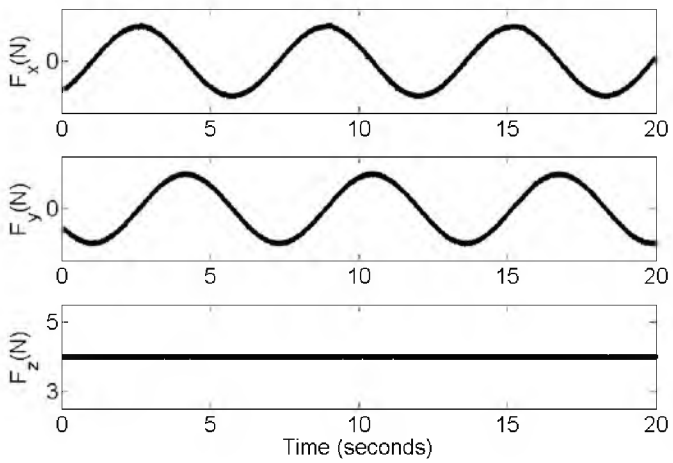

Fig. 2.14. Tracking behavior on rotating shear force. A constant normal force is held while a constant-magnitude shear force rotates with a frequency of 1 $\frac{\text{rad}}{\text{s}}$. The forces in all three directions exhibit errors less than 0.05 N. The desired angle is held with an error less than 0.1 rad.

dependent upon the choice of force commands. If the shear force is too large relative to the normal force, the finger and flotor will experience relative motion. Once this relative motion is introduced, it has been observed that the system quickly becomes unstable and oscillates until it moves outside the desired workspace, requiring that the system be reset. Thus, when designing trajectories, it is necessary to include only those combinations of shear and normal force that prevent slip between the finger and flotor. Experience has shown that a static friction coefficient of $\mu = \frac{2}{3}$ yields target force combinations without slip on all fingers when using the equipment described above.

2.5.2 Trajectory Tracking

Each of the 17 subjects experiences 108 individual data collection periods. These periods last from 10 s to 60 s with normal forces varying from 0 N to 6 N and shear forces varying from -4 N to $+4$ N. The flotor is commanded to hold the rotation angles at identically zero. A sample data set is shown in Fig. 2.15. This set exhibits 0.06 N RMS error in the x- and y-directions and 0.03 N in the z-direction, with 0.03° RMS error in θ_x, 0.01° in θ_y, and 0.1° in θ_z.

Fig. 2.16 shows the force RMS errors across all subjects, grouped by LED color and force direction. It is to be expected that no significant difference exists between the two LED colors, as this should have no effect on the controller or the subject's performance. Although each force direction is slightly different, both LED colors exhibit approximately the same error (White: 0.057 ± 0.001 N, Green: 0.058 ± 0.002 N).

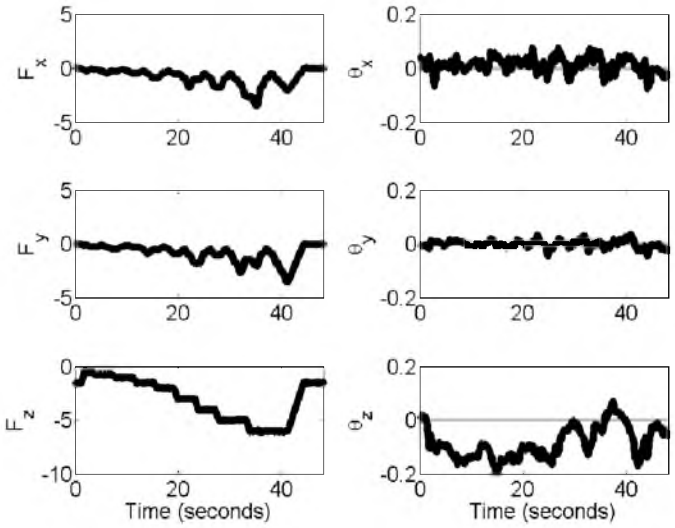

Fig. 2.15. Measured forces and angles during an experiment. The forces are plotted in Newtons, with the angles in degrees. The data in this experiment show 0.06 N RMS error in F_x and F_y, with 0.03 N in F_z. The angular errors are 0.03° in θ_x, 0.01° in θ_y, and 0.1° in θ_z.

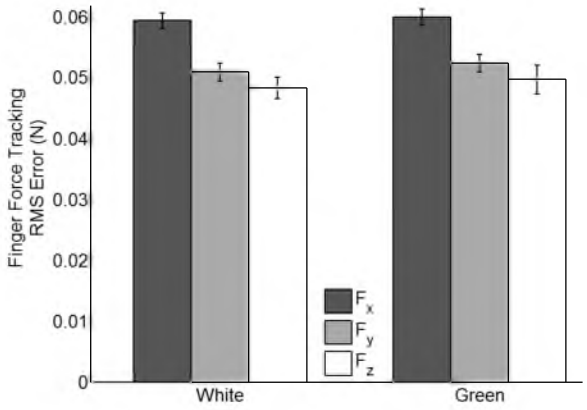

Fig. 2.16. Measured vs. desired force-tracking RMS error, grouped by LED color and force direction. As expected, no significant difference exists between the LED colors, in spite of differences among the force directions.

The force RMS errors grouped by trajectory type (Cartesian vs. Cylindrical) and force direction are shown in Fig. 2.17. The choice of trajectory has a significant effect on force tracking RMS error, with the Cartesian grid having a significantly larger error (0.061 ± 0.002 N) than the Cylindrical grid (0.054 ± 0.001 N). Not only do the two grid designs exhibit significantly different RMS tracking errors, but the pattern of error between the two is different as well: the Cartesian grid has significantly larger error in both the x- and y-directions, with no statistically-significant difference between the two in the z-direction.

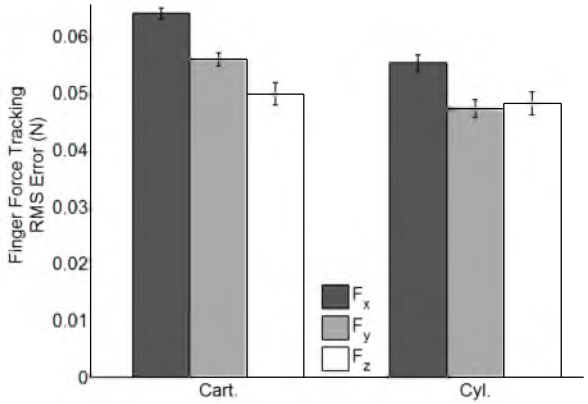

Fig. 2.17. Measured vs. desired force-tracking RMS error, grouped by calibration grid and force direction. The Cartesian grid exhibits significantly larger error than the cylindrical grid. The patterns among the force directions are also significantly different.

The force RMS errors are aggregated by finger and force direction in Fig. 2.18. The difference in the way force direction interacts with the choice of finger in its effect on tracking error is shown clearly here. When acting on the index finger, all three force directions exhibit tracking errors that are approximately equal (0.054 ± 0.001 N). When the flotor acts on the thumb, a significantly larger tracking error is detected in F_x (0.064 ± 0.001 N) than either F_y (0.055 ± 0.001 N) or F_z (0.053 ± 0.001 N). Recall that the contact condition is different when calibrating the thumb. This is most likely the cause of the difference between the two fingers. It may be possible to confirm this by calibrating other fingers using the same contact condition as the index finger.

In Fig. 2.19, the angle-tracking RMS errors are grouped by LED Color and angle

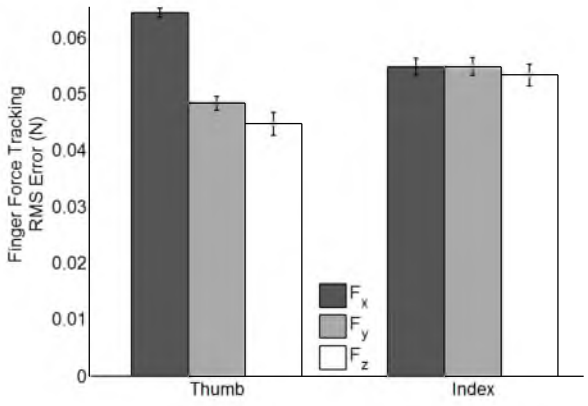

Fig. 2.18. Measured vs. desired force-tracking RMS error, grouped by finger and force direction. The force directions clearly interact with the choice of finger in their effect on tracking error. All force directions exhibit approximately the same error on the index finger, while the three force directions have significantly different effects on the thumb.

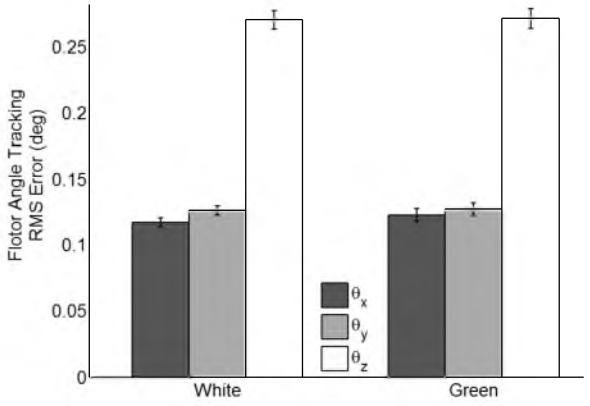

Fig. 2.19. Measured vs. desired angle-tracking RMS error, grouped by LED color and angle direction. As expected, no significant difference exists between the two LED colors, in spite of differences among the angle directions.

direction. As with the forces, the LED Color should have no effect on the accuracy of the angular position controller. There is no statistically significant difference in the angular RMS error in any direction between the two LED colors, as expected.

Fig. 2.20 shows the angle-tracking RMS errors grouped by calibration grid and angle direction. The angular error is significantly lower in all three directions for the Cylindrical grid than for the Cartesian grid. An additional significant effect is detected: For the Cartesian grid, the errors in θ_x ($0.135 \pm 0.004^\circ$) and θ_y ($0.137 \pm 0.004^\circ$) errors are statistically similar. For the Cylindrical grid, however, these two errors are significantly different, with θ_x having an RMS angular error of $0.104 \pm 0.004^\circ$ and θ_y having an RMS angular error of

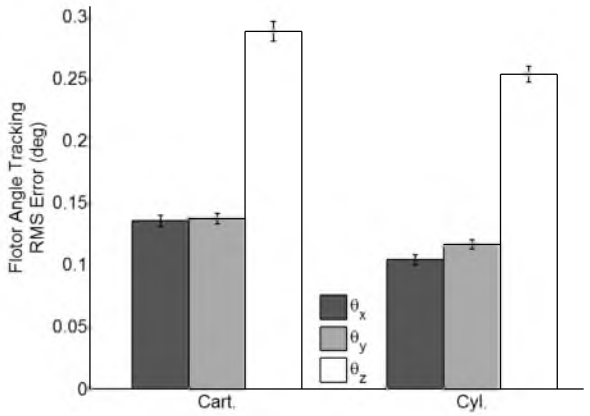

Fig. 2.20. Measured vs. desired angle-tracking RMS error, grouped by calibration grid and angle direction. The tracking error is significantly lower in all three directions for the Cylindrical grid than for the Cartesian grid. In addition, the errors in θ_x and θ_y are statistically different for the Cylindrical grid, while for the Cartesian grid, they are similar.

$0.116 \pm 0.004^{\circ}$.

The angle-tracking RMS errors, grouped by finger and angle direction, are shown in Fig. 2.21. The interaction effect is less pronounced than it was in the force tracking results, although it is still significant. The angular error is significantly larger in both the x- and z-directions regardless of the finger, while in the y-direction the reverse is true. As with the force-tracking differences, such differences are most likely due to the differences in contact condition between the two fingers.

Note, however, that all of these angular errors are less than one-third of one degree, or approximately 2 % of the full range of $\pm 8^{\circ}$ of rotational movement in any direction. Even a statistically significant difference between rotational directions or fingers is not likely to make much of a difference in stability during a calibration test.

2.5.3 Model Calibration - Force Prediction

The end goal for this system, of course, is to improve the process of developing a model relating finger images to fingerpad force. The EigenNail Magnitude Model, introduced previously [36], is a Principal Component Regression model. It relates the coordinates of the image in the eigenspace spanned by the eigenvectors of the pixel intensity values to the forces on the fingerpad. The prediction model takes the form

$$\mathbf{p} = \bar{\mathbf{p}} + \mathbf{\Sigma}_p \mathbf{b}_p \tag{2.9}$$

$$\mathbf{F}_s = \mathbf{A}\mathbf{b}_p + \mathbf{f}_0 \tag{2.10}$$

where $\mathbf{p}$ is a vector of pixel intensities, $\bar{\mathbf{p}}$ is the mean pixel vector, $\mathbf{\Sigma}_p = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_k \end{bmatrix}$ is the matrix of EigenNails, $\mathbf{A} = \{a_{ji}\}$ is a matrix of coefficients, $\mathbf{b}_p = \begin{bmatrix} b_{1p} & b_{2p} & \cdots & b_{kp} \end{bmatrix}^T$

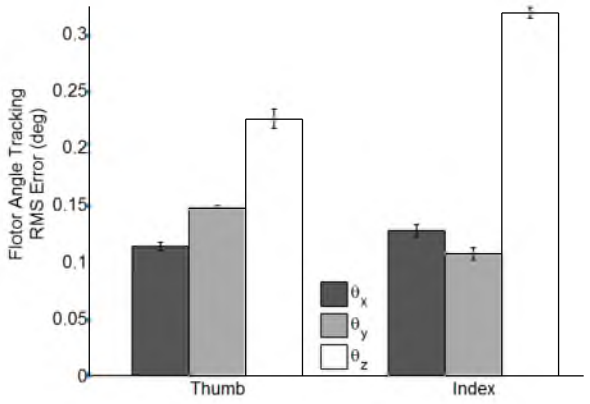

Fig. 2.21. Measured vs. desired angle-tracking RMS error, grouped by finger and angle direction. The tracking errors between the fingers show a complicated interrelationship here, with significance being detected in nearly every pairing.

is the vector of Nail Space coordinates and $\mathbf{f}_0$ is the offset force vector.

Fig. 2.22 shows the force prediction RMS error grouped by force direction for a series of validation experiments. In these experiments, 25% of the images for a given subject/finger/LED color combination were set aside as "validation" data, while the rest were used to form the model. Then, the model was applied to the validation data, and the resulting predicted force was compared to the measured force. It may be seen from the figure that $\mathbf{b}_p$ correlates well with the finger forces $\mathbf{F}_s$, resulting in RMS errors of $0.56 \pm 0.03°$ in F_z, with $0.54 \pm 0.02°$ in F_x and $0.56 \pm 0.03°$ in F_y. The choice of force direction has no significant effect on the force prediction RMS error.

A secondary, qualitative check that may be performed is to create a set of synthetic images where each pixel's intensity indicates the contribution of that pixel in the prediction model to the corresponding direction of force. The formula used to generate these images is

$$\mathbf{p}_j = \sum_{i=1}^{k} a_{ji} \boldsymbol{\phi}_i \sqrt{\lambda_i} \tag{2.11}$$

where λ_i is the eigenvalue corresponding to the ith EigenNail, and the index j corresponds to the force direction (x, y, and z). The resulting images for one subject are shown in Fig. 2.23 (index finger) and Fig. 2.24 (thumb). The patterns shown here correspond to those expected: the asymmetric pattern exhibited when the finger is under x-direction force, the opposing bands near the distal end and across the middle of the nail under y-direction force, and the band near the distal end of the finger paired with the discoloration in the finger along the sides of the nail under z-direction force all appear in the corresponding image

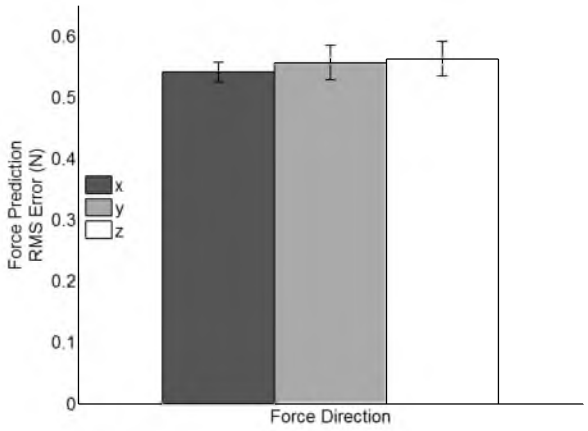

Fig. 2.22. Force prediction RMS error, grouped by force direction. The RMS error in F_z is $0.56 \pm 0.03°$, with slightly lower values in the two shear force directions. No statistically significant effects are detected.

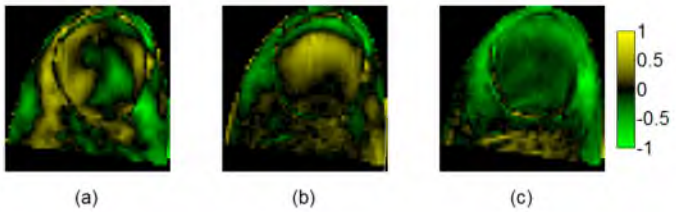

Fig. 2.23. Qualitative prediction model analysis (index). The intensity of each pixel in these images represents the contribution made by that pixel to the force prediction in each of the (a) x-, (b) y-, and (c) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$, with yellow having a more positive effect and green having a more negative effect. The asymmetric pattern in (a) matches the patterns visible on the thumb when x-direction force is applied, while the alternating green and yellow bands across the distal end and middle of the nail in (b) correspond to the patterns visible when y-direction force is experienced. In (c), the brightest regions correspond to those that become most white when applying z-direction force.

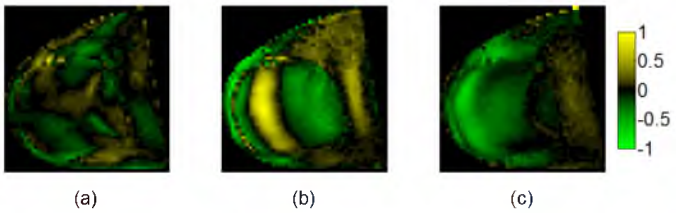

Fig. 2.24. Qualitative prediction model analysis (thumb). The intensity of each pixel in these images represents the contribution made by that pixel to the force prediction in each of the (a) x-, (b) y-, and (c) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$, with yellow having a more positive effect and green having a more negative effect. The asymmetric pattern in (a) matches the patterns visible on the thumb when x-direction force is applied, while the alternating green and yellow bands across the distal end and middle of the nail in (b) correspond to the patterns visible when y-direction force is experienced. In (c), the brightest regions correspond to those that become most white when applying z-direction force.

maps.

2.5.4 Calibration Grids

Finally, the effect of the size of the data set on the force prediction error is investigated. The intent of this experiment is to determine how many images must be collected to achieve the prediction error shown previously. Thus, the number of images is reduced and the data set is divided into calibration and verification sets. Then, the calibration set is used to form the model, and the verification set is used to validate the model. This experiment is repeated 100 times for each combination of subject, finger, and LED color, with a different randomly-

selected division each time. The results are shown in Fig. 2.25. When the number of images is reduced to 625, the error (0.59 ± 0.02 N) is not statistically different from the standard model (0.55 ± 0.02 N). When the number of images is reduced to 500 (0.68 ± 0.02 N), a statistically significant difference is detected. The 250-point model has a much larger error (1.00 ± 0.04 N).

2.6 Conclusion

This paper has demonstrated the development and tuning of a new force controller for interacting with the human fingerpad, with the end goal of calibrating a new method for transducing force on the fingerpad by measuring the coloration of the fingernail and surrounding skin. The controller has a step response with an overshoot of less than 30 % and a 2 % settling time of less than 0.13 s in all three directions of force. The three directions of force exhibit an RMS tracking error of 0.065 ± 0.002 N normal force and 0.054 ± 0.001 N shear force. The three angular directions have an RMS tracking error of $0.173 \pm 0.003°$.

The automatically calibrated force prediction model predicts force simultaneously in all three directions, with an RMS validation error of 0.56 ± 0.03 N in all directions. The shear force is estimated within 8 % of the full range of -4 N to $+4$ N, while the normal force is estimated within 10 % of the full range of 0 N to 6 N. In addition to correctly predicting forces, the new model presented here is shown qualitatively to be dependent on the correct color patterns and regions of the finger. Past force prediction models sometimes relied on movement of the finger relative to the frame of the image. This qualitative technique may be used in future work to assess the accuracy of registration and prediction models.

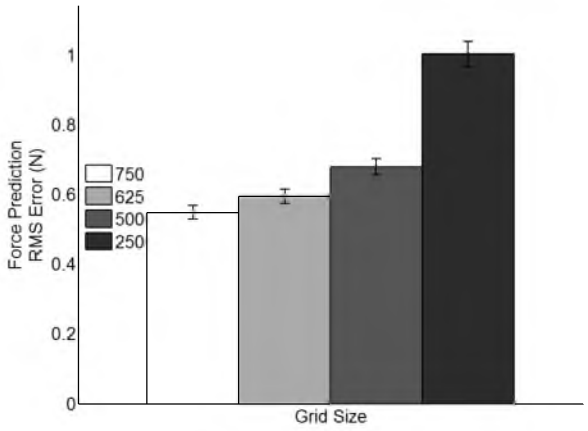

Fig. 2.25. Force prediction error by number of images. The standard data set contains 750 images and is reduced according to the numbers shown. While the 625-point data set does not result in a statistically-significant difference in error, the other sizes do.

The authors intend to use these results in the future to perform a study of human grasping, demonstrating that the force in each finger may be estimated reliably using the imaging method. This approach would greatly improve existing grasping research, as it would no longer be necessary to instrument either the object or the finger. It would also allow the object to be grasped in a natural way, rather than requiring the test subject to place the fingers on prespecified contact points designed for measuring the force.

CHAPTER 3

IMAGE REGISTRATION USING ACTIVE APPEARANCE MODELS FOR CALIBRATION OF FINGERNAIL IMAGING

3.1 Abstract

This paper demonstrates the registration of finger images using Active Appearance Models (AAM) for the purpose of predicting fingerpad force while also presenting an investigation into various AAM parameters. It is shown that these individually calibrated models may be used over a wide variety of subjects and that simpler models preserve both the registration and force-prediction accuracy of the more complex models. The mean registration error is shown to be 1.5 ± 0.3 (RMS pixel intensity error), less than $0.5\,\%$ of the range of 0 to 255, while the corresponding force prediction error is found to be $0.55\,\mathrm{N}$, $9.1\,\%$ of the full range of forces measured.

The Standard model that achieves these results uses 75 landmark points across the nail and finger and between 9 and 10 training images, calibrating the AAM using the entire RGB color space. The remaining images, between 580 and 750 per subject, are then registered using the Search Model formed as part of the AAM. The reduced models achieve statistically similar results while reducing the number of landmark points to 25 and processing only the Green color channel of the images. Since these reductions all result in fewer pixels and fewer landmark points to calculate when performing the registration and the force prediction, the images may be processed more quickly.

3.2 Introduction

Fingernail imaging has been shown to be an effective method of measuring contact force on the human fingerpad [1–3, 27]. Using fingernail imaging, these contact forces may be measured without restricting the haptic sense or requiring that force sensors be placed

in specific contact locations. This method of fingerpad force measurement can improve human/machine communication, for example in rehabilitative environments or machine learning opportunities. As patients interact with robots designed to aid rehabilitation, the finger forces may be measured using imaging rather than costly force sensors, also allowing patients more freedom in contact locations. When training robots using supervised learning, a vision system may measure finger forces using fingernail imaging, allowing a wider range of interactions for the system to learn.

For fingernail imaging, contact forces are measured by inspecting the coloration of the tissue beneath the fingernail. The coloration change is caused by the constriction of blood vessels in the flesh of the finger. Nearly transparent, the fingernail transmits the color change from these underlying tissues, allowing the imaging of fingernails and the surrounding skin. The effect is common across a wide range of people [2]. This paper discusses the application of Active Appearance Models (AAM) as a registration method for fingernail images. While previous registration methods are generally effective, each is deficient in some aspect, preventing accurate application of calibration models to data recorded in another setting.

Fingernail imaging as a method for predicting force is a two-step process. First, a calibration phase is required. This includes the collection of images while the finger experiences a series of forces designed to cover a desired force space. Once the calibration images have been collected, a prediction model relating the intensity values to the forces is created. Second, this force prediction model is applied to the intensity values of an image to estimate the force. If the actual finger forces are measured, they may be compared to the predicted force to determine the model's accuracy. Models have been developed that estimate a single direction of force magnitude [3], the most likely force direction [27], and the full 3-D force magnitude [26].

Using the fingernail coloration effect to detect finger force has many applications. A computer mouse may be replaced with a vision system that senses finger force readings and converts them to cursor velocities and clicks. Another application is to measure finger forces during grasping. A problem with traditional grasping studies is that they typically require a change of grasp to achieve measurement. When the fingers are instrumented [7,73], the potential exists for loss of haptic sense. When the object is instrumented [9, 74], the finger contact points are constrained. Although it is possible to perform unconstrained grasping with instrumented objects [8], finger contact locations are still somewhat limited. As a solution, cameras could be placed to record the coloration effect of the fingers. This

approach could lead to a more natural grasp of the object and reduce the amount of time required to set up such experiments.

3.2.1 Prior Registration Methods

Proper image registration is necessary to ensure the same locations of the finger will be compared from one image to the next. Prior methods used for fingernail image registration include: 2D-to-3D Registration, the Harris Method, the Canny Method, and the Scaled Rigid-Body Transform. Each method has its drawbacks.

The *2D-to-3D Registration method* [3] uses a stereo image of the finger to generate a 3-D model of the finger. 2-D images are registered to this model using fiducial marks that are drawn on the finger. While this method has been seen to be accurate, the complexity of the model requires longer processing time. All other techniques assume the finger to be a flat plane and attempt to register the finger to some template image.

The *Harris method* [26, 27] correlates the Harris feature points [28] from an image to be registered with feature points in the template image, based on pixel intensities in the surrounding neighborhoods. RANSAC [29] is then used to find the transformation that best fits the correlations. However, this approach fails to register some images due to the lack of well-defined feature points on the finger. It also frequently identifies correlations between points that do not correspond, resulting in skewed images that are improperly registered. Additionally, since the RANSAC algorithm is probabilistic, the method is not repeatable. It has correctly registered an image on one attempt and failed to do so on another simply because of the random number generator.

The *Canny method* [26] uses Canny edge-finding [30] to locate the edge points in the image. As with the Harris Method, these points are then correlated between the image and the template image, and RANSAC is used to find the correlated pairs that produce a minimum-error transformation. This method has been more successful at registering images, but has proven to be slower. It also requires the optimal values of the parameters (i.e., the high and low thresholds and the standard deviation of the Gaussian) to be experimentally determined for each data set, based on the lighting conditions as well as the shape of the finger itself. In addition, images where the finger does not continue to the edge of the image have a false edge that should not be used to register the image.

The *Scaled Rigid-Body Transform* [26, 31] first locates the finger in the image using thresholding and connected-component (CC) analysis. The rotational offset of the finger is estimated from the major axis of this CC. The finger is rotated to align with the vertical axis and the image is cropped to the edge of the CC. Although this method is fast and accurate

for predicting force, it does not accurately register the images. The seemingly accurate results previously demonstrated in force prediction [26] are an effect of the movement of the finger within the bounding-box of the final image, rather than the changing intensity of the finger regions themselves. In other words, the model was being calibrated to the movement of the finger within the bounding-box in addition to the coloration of the finger. This problem is due to the basic assumption of this model, that the motion of the finger was limited to (1) motion in and out of the image plane (i.e., scaling) and (2) motion parallel to the image plane (i.e., x-y translations and z-rotation). While experience has shown that rotations out of the plane (i.e., x-rotation and y-rotation) are negligible, one other modality is ignored: the finger may deform as force is applied, leading to apparent motion between the nail and the edge of the finger.

The *Hand-to-Finger Transform* [32] finds the hand in an image using a predefined skin color vector. The angle between each pixel intensity and this color vector is calculated, and if the angle is less than a given threshold, the pixel is defined to be part of the hand. The fingertips are found using Integral Image [33] to search for hand pixels whose neighborhood contains more than a given threshold of other hand pixels. The finger direction is estimated by searching for the chord of a circle at the centroid of the fingertip that crosses the hand pixels. The finger angle is estimated to be the same as the angle of the radial line that bisects the chord. While this method appears to be successful on all images, its underlying assumption is that the image contains the entire hand. When images contain only the finger down to the PIP joint, the results are mixed. In addition, several parameters must be adjusted for any given data set, depending on the skin's color and the lighting conditions. Finally, once the fingers have been located, the rest of the procedure is the same as for the Scaled Rigid-Body Transform, meaning that this method ignores skin deformation.

3.2.2 Active Appearance Models

A novel registration method has been developed [36, 37] that iteratively uses Active Appearance Models (AAM) to register all of an individual's data, compensating for scaling, translation, and rotation. AAM have been used to register images of faces [38–41], electrical components [38], and medical images (including knee MRI [38, 42], cardiac MRI [40, 43], dental radiographs [44–46], vertebra MRI [47], eye OCT scans [48], abdominal organs [49], and red blood cells [50]). In addition, AAM have been applied to count people [51], to perform expression recognition [52, 53], to synthesize speech [54], and to synthetically age face images [55].

An AAM is formed in effectively the same way, no matter the object to be found

(Fig. 3.1). The AAM technique requires a group of training images, which should be representative of the modes of variation of the data set. (This requires some knowledge of the underlying characteristics of the data.) A set of common landmark points are identified for each of the training images. A statistical model of Shape variation is generated by aligning each set of points to a common frame of reference and applying a Principal Component Analysis (PCA) to find the modes of Shape variation. Each training image is then warped to the mean shape, and the gray-level information is extracted. These values are normalized, and PCA is applied to find the modes of Texture variation. The Shape and Texture parameters for each training image are then combined into a single Appearance vector, and PCA is applied a third time. This generates the modes of Appearance variation. Finally, a Search Model is created by correlating the pixel intensities to the Appearance and pose parameters. Given a new image and an initial estimate of the pose parameters of the mean shape, this Search Model may be used to determine the appropriate Shape, Texture, and Appearance parameters, as well as the position, scale, and orientation (Fig. 3.2). Thus, it may be used to register the image to a template, such as the mean shape.

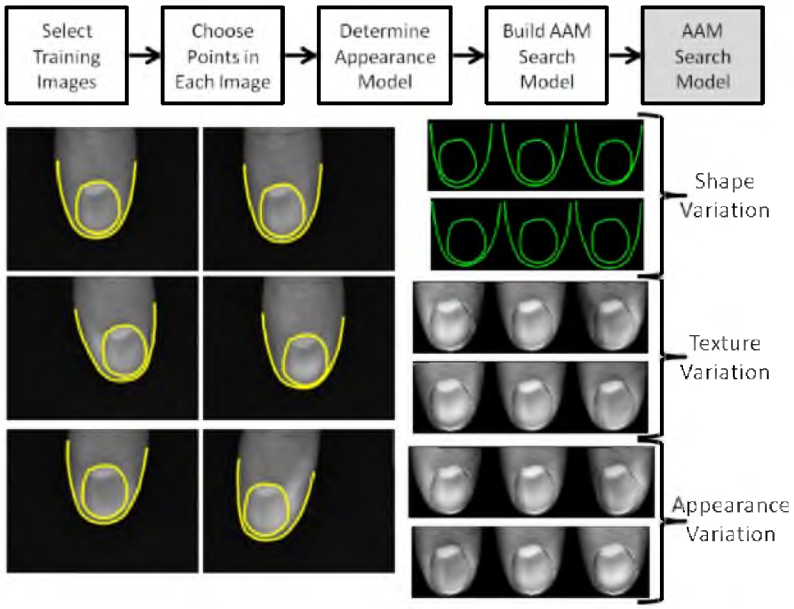

Fig. 3.1. Forming an AAM from fingernail images. The training images are chosen. For each of these images, the landmark points are selected. Using the method of [38], the AAM is formed, including the Shape, Texture, and Appearance Variation, and the Search Model. For the subject shown here, six training images were selected, representing F_{zero}, $+F_x$, $-F_x$, $+F_y$, $-F_y$, and F_z. The first two modes each of Shape, Texture (gray-level), and Appearance (combined Shape and Texture) variation are shown.

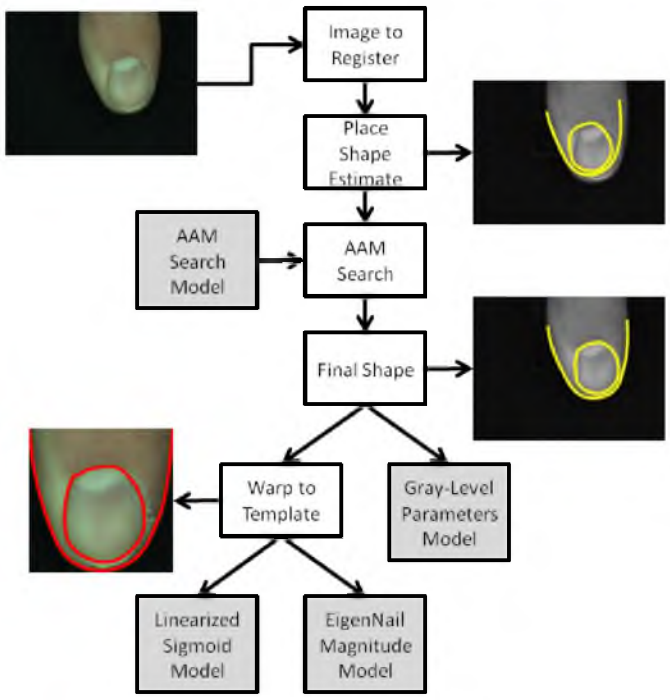

Fig. 3.2. Image registration procedure using the AAM search model. The mean shape is placed in an estimated location over the image to be registered. The AAM Search Model is applied to determine the final shape. The position of the points in this final shape is used to warp the image to the template image. The parameters used to make any of the three force prediction models may then be extracted.

3.2.3 Current Work

This paper details the application of AAM to fingernail images, a process which began with the publication of [36, 37]. Novel contributions of this work are numerous. First, an enlarged data set improves the statistical significance of the results and demonstrates the applicability of the method over a larger range of the general population. Second, the method for generating landmark points has been refined, leading to improved registration. Third, a new set of contours is devised and examined to attempt to reduce the remaining residual. Fourth, a more in-depth comparison to previous registration and prediction methods is presented. Fifth, the iterative technique used to refine the training data set is explained. Finally, a novel method for estimating registration error is presented due to the absence of true comparisons.

The paper is organized as follows: In Section 3.3, the experimental procedures are explained, while the registration method is detailed in Section 3.4. The image registration error is estimated and analyzed in Section 3.5. Various modifications to the registration method are discussed in Section 3.6, and the registration error of these modifications is

examined. Finally, in Section 3.7, the registered images are used to predict force, and the results are discussed.

3.3 Experimental Procedures

Data are collected using an automated calibration platform [26] that applies forces to the finger while the subject remains relatively passive. Forces are recorded using an ATI Nano17 6-axis Force Sensor, with a range of 17 N normal force, 12 N shear, and 120 N · mm torque. The desired levels of force are visited at a rate of approximately one every 0.5 s to ensure that the finger coloration has time to settle [3]. To reduce the number of files that need to be coordinated, the force is encoded into the $(1, 1)$ pixel of the corresponding image. This results in an effective resolution of 0.047 N [26]. A piece of two-sided tape is placed between the end effector and the fingerpad to ensure nonsliding contact.

Images are collected using a Point Grey Research FLEA 1.0 camera, recording one RGB image at each level of desired force, at a resolution of 1024×768 pixels. The camera is placed approximately 30 cm above the finger so that the region of interest of the finger occupies approximately 300×300 pixels of the image. The camera is located such that the finger is nearly centered in the image. Thus, when the finger is displaced during calibration, it remains within view of the camera. The camera gain, exposure time, and shutter speed are all maintained at a constant level to provide consistency across images and subjects.

Data from 16 subjects is collected, including images of both thumb and index finger. Between 550 and 750 images from each subject/finger combination are recorded, representing combinations of normal forces between 0 N to 6 N and shear forces in the range -4 N to 4 N. As explained in Section 3.4, an AAM is formed using between 9 and 10 training images for each of the 32 subject/finger combinations and used to register the remaining images. Once all of the images for one data set have been registered, they may be used to form a force prediction model. The calibration apparatus for both thumb and index finger is shown in Fig. 3.3.

3.4 Registration Procedure

Once the images have been collected, they must be registered. As explained in Section 3.2, each of the previous image registration methods has some fault or limitation that renders it unsuitable or, at best, undesireable. A novel registration method has been developed [36, 37] that iteratively uses AAM to register all of an individual's data. As summarized in Fig. 3.1 and Fig. 3.2, this registration method consists of (1) selecting the training images, (2) choosing landmark points within those images, (3) forming the Shape,

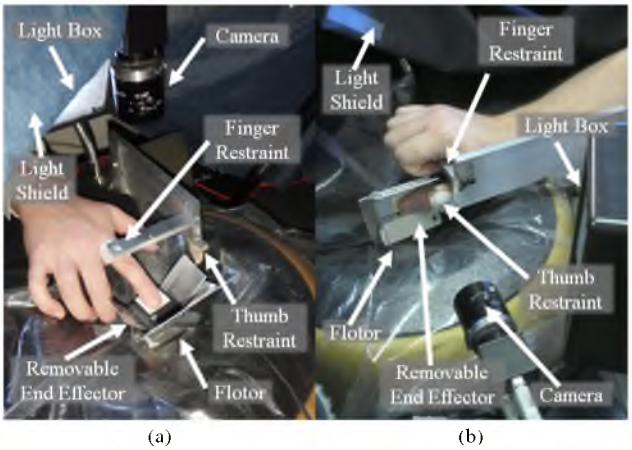

Fig. 3.3. Experimental apparatus. (a) Calibrating the index finger. The finger rests between the restraint and the end effector while the sensor, mounted on the flotor, records forces. A light box is placed above and behind the finger to provide consistent illumination while a shield blocks ambient light. The camera is stationed directly above the nail. (b) Calibrating the thumb. Due to the differing kinematics of the thumb, its calibration takes place at 90° to the finger. The restraint and end effector are different, as are the position of the camera and light source. The calibration apparatus compensates automatically for the differences in required forces.

Texture, Appearance, and Search Models, (4) registering all of the other images using the Search Model, and finally (5) refining the training set and the subsequent models, if needed. This technique has two additional benefits: the modes of variation of the finger under normal and shear forces are identified, and new sets of parameters are determined that can be incorporated into force-prediction models. These benefits will be explored at appropriate locations.

3.4.1 Training Images

First, a subset of the data set must be selected and identified as training images. These should, within reason, exhibit the full variation of Shape and Texture present in the data set [39]. Some past fingernail imaging data sets, such as those collected for the original EigenNail experiments [27], consist of several groups of images recorded at approximately the same levels of force. The desired force in the x- and y-directions for one such data set are plotted in Fig. 3.4(a). In this data set, it is reasonable to suppose that a single image from each of these groups would suffice to form a training set as it may be expected that all images within each group have approximately the same characteristics.

The goal of the current work is to characterize variation of the finger coloration over the full range of forces, rather than classify directional patterns of coloration. Because of this,

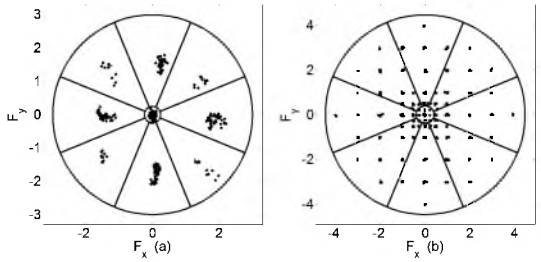

Fig. 3.4. Force grouping of finger images. (a) Shear forces from a directionally grouped data set, where many images are collected at a few specific force levels. Such a set is easily divided into groups. (b) Shear forces from a more continuously distributed data set, where only one or a few images are collected at many specific force levels. Dividing such a data set into groups is not as intuitive, although the same general procedure may be used. In both cases, forces near zero shear form one group. Other forces are grouped according to shear force angle. Lines in both images indicate divisions between groups.

the images in experiments conducted for the current work are not grouped as conveniently, but are instead more continuously spread throughout the force space, as illustrated in Fig. 3.4(b). To attempt to obtain a representative set of training images, however, a similar principle may be applied. The images are grouped by location in the force space, first by their proximity to the z-axis and then by their angular coordinate. The groups are shown in Fig. 3.4(b). A representative image from each group is randomly selected, and together these images form the set of training images.

3.4.2 Landmark Points

Once the training images have been selected, the landmark points must be identified. At first glance, the human finger appears to contain several well-defined landmarks, as shown in Fig. 3.5(a). The endpoints along the edge of the finger (pair A), the points where the lunula edge (the half-moon shape at the proximal end of the nail) meets the edge of the fingernail (pair B), or the points where the free matrix (the white portion that is customarily trimmed) terminates on either side of the fingernail (pair C) might each be chosen.

However, there are potential problems with each of these landmark points. If Pair A were selected to lie at the edge of the images in Fig. 3.5(b) and (c), the Shape Model would interpret that finger (b) is longer than finger (c). Alternatively, if an arbitrary point along the finger were chosen, one could imagine the individual placing the point having difficulty choosing the location reliably. In Fig. 3.5(c) and (e), the lunula is either missing or obscured. The landmark points illustrated by pair B could not be reliably placed. The problem with pair C is illustrated in all of the images in Fig. 3.5(b) through (e). Each of these fingers is trimmed differently, or the shape of the nail is different. The placement of pair C would

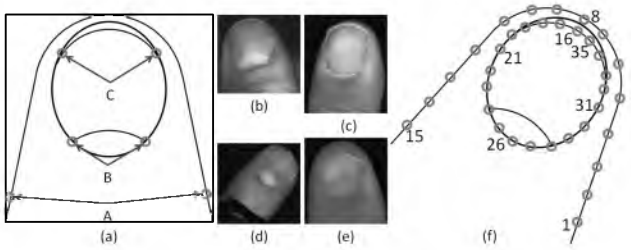

Fig. 3.5. Landmark point location considerations with the human finger. (a) Potential landmark points for AAM on the human finger are shown. The endpoints for the finger edge contour (Pair A), the points where the lunula edge meets the edge of the fingernail (Pair B), and the points where the free matrix terminates on either side (Pair C) are suggested. Each pair creates problems in certain cases. The finger images may contain differing lengths of the finger, as in (b) and (c), which would inhibit the proper placement of Pair A. The lunula is not visible in some fingers, as in (c) and (e), making Pair B unsuitable. The variation in nail-trimming condition and nail shape shown in these images indicate that Pair C would likewise make a poor landmark. In (f), two contours are used to represent the finger: the outer edge (points 1 15) and the fingernail edge (points 16 35).

vary depending on the judgment of the individual placing the points. Even if these three sets of points could be selected reliably, they would be insufficient for registering the full set of finger images. Other landmarks might be chosen that are specific to the individual, such as wrinkles in the skin, hangnails, or other imperfections, but these would not be generally applicable from one subject to the next. Since a registration method that can be generalized is desired, a different approach is required.

Rather than locating specific landmarks, two contours are chosen to represent the variation in shape of the finger, as illustrated in Fig. 3.5(f). The finger contour represents the outside edge of the finger. The nail contour follows the edge of the fingernail, tracing along the inside edge of the free matrix to eliminate differences in nail trimming. Points are selected by hand along each of the two contours.

The following algorithm is then applied to smooth the data:

1. Smooth each contour using linear interpolation.
2. Estimate the angle between the finger vector and the positive x-axis.
3. Identify the most distal point on the fingernail contour (along the finger vector).
4. Starting with this point, smooth the nail contour using linear interpolation.
5. Calculate the fingernail length (L_n) (along the finger vector).
6. Identify the most distal point on the finger contour (along the finger vector).
7. Starting from this point, the endpoints of the finger contour are found at $1.5L_n$.
8. Smooth the revised finger contour using linear interpolation.

These steps are described in detail in the following sections. For the standard registration

model used in this paper, 30 points are used to form the final finger edge contour, and 45 points are used to form the final fingernail edge contour. The final smoothed contours for each image are shown in Fig. 3.6 and Fig. 3.7.

Smoothing of the contours by interpolation is illustrated in Fig. 3.8(a). The finger contour is interpolated such that the x- and y-coordinates are each spaced evenly between the first and last points. The nail contour, on the other hand, is arranged such that the first point is repeated at the end. This allows the interpolation to take place over the entire intended loop. One more point than required is generated since the first and last points of the output will also be identical. This repeated final point is eliminated after the interpolation. Initially, each of the two contours is interpolated so that they contain 51 points to standardize the following steps.

3.4.2.1 Finding the Finger Angle

Determining the approximate angle of the finger involves finding the average of the angles of each of the two sides, as shown in Fig. 3.8(b). The first third of the finger contour

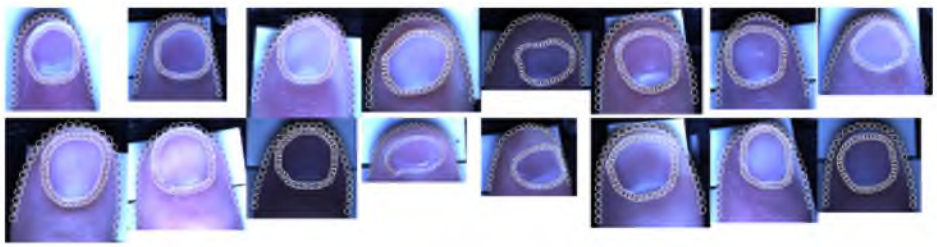

Fig. 3.6. Index finger training images from all subjects. The automatically smoothed contours are shown on the images. Due to the design of the calibration equipment, index finger images always point up.

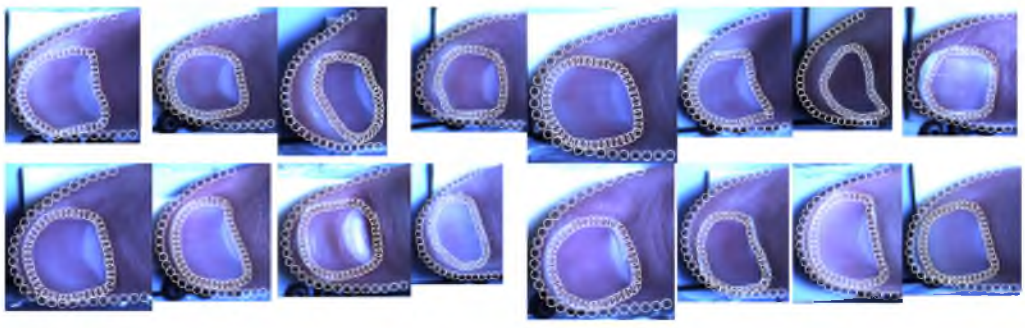

Fig. 3.7. Thumb training images from all subjects. The automatically smoothed contours are shown on the images. Due to the design of the calibration equipment, thumb images always point to the left.

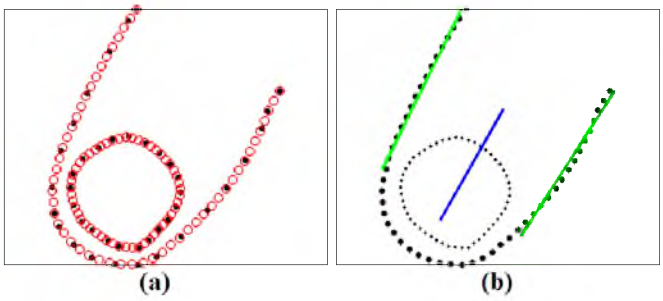

Fig. 3.8. Smoothing contours/finding the finger angle. (a) Smoothing a sample finger contour by linear interpolation. The black dots are the initial (manually-selected) points, while the red circles are the interpolated points along the contours after the first step. (b) The angle that the finger direction vector (blue line) makes with the positive x-axis is approximated by finding the average of the angles of each of the two sides (green lines).

is designated as the "right" side of the finger. The angle (θ_r) between these points and the positive x-axis is estimated. (All angles in this paper are measured positive clockwise since they are in image plane coordinates where the x-axis is positive right and the y-axis is positive down.) Likewise, the final third of the finger contour is designated as the "left" side of the finger, and its angle (θ_l) relative to the x-axis is estimated. These two angles are averaged to estimate the finger angle, $\theta_f = 0.5\,(\theta_r + \theta_l)$. The finger angle may also be used to define the finger direction vector, $\hat{\mathbf{v}}_f = \begin{bmatrix}\cos\theta_f & \sin\theta_f\end{bmatrix}^T$.

Since the two-argument arctangent function is used to calculate the side angles, a range correction may be necessary. Given that the range of the `atan2()` function (in Matlab) is $[-\pi, \pi]$, if the finger is pointing approximately to the left, the two line angles may be at opposite ends of the range. Averaging such values would result in a finger angle near zero (i.e., pointing to the right). To correct for such an event, the angles are compared prior to averaging. If their difference is more than 1.5π radians, the positive angle is reduced by 2π before averaging.

3.4.2.2 Adjusting the Nail Contour

To adjust the nail contour, it is first desired to find the most distal point on the nail contour, along the line through the nail centroid and parallel to the finger direction vector, as illustrated in Fig. 3.9(a). To do this, the nail points are translated so their centroid lies at the origin and then rotated into a coordinate system defined such that the new x-axis is parallel to $\hat{\mathbf{v}}_f$, with the same z-axis as the original image. In terms of a coordinate transformation, this may be written as

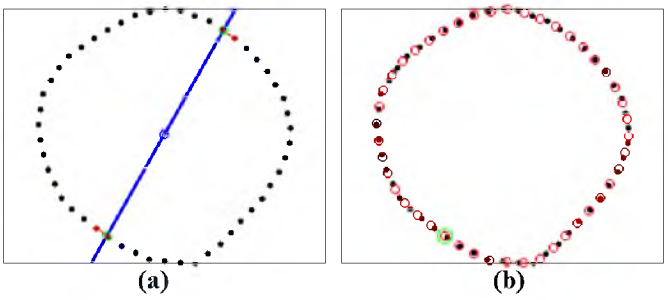

Fig. 3.9. Adjusting the nail contour. (a) Finding the most distal point of the nail contour. The finger axis is shown through the nail centroid, and the two points where it intersects the nail contour are located. The one with the larger (more positive) coordinate parallel to the finger axis is chosen as the most distal point. (b) Smoothed nail contour (red circles), with initial point (larger green circle), superimposed over original points (black dots).

$$\begin{bmatrix} x_{\parallel} \\ x_{\perp} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_f & -\sin\theta_f & -\bar{x} \\ \sin\theta_f & \cos\theta_f & -\bar{y} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3.1}$$

where $\bar{x} = \frac{1}{N}\sum x_i$ and $\bar{y} = \frac{1}{N}\sum y_i$ are the centroid of the n nail points. The zero crossings of the perpendicular coordinate ($x_{\perp}$) are then found by locating the two sets of consecutive pairs of indices where the sign of the perpendicular coordinate changes. The initial and final points are considered a consecutive pair for the purposes of this calculation.

The most distal zero crossing is found by comparing the signs of the parallel coordinate and selecting the positive one. The exact zero crossing location is then estimated by linear interpolation between these points. Using this new starting point, the entire contour is smoothed again using linear interpolation, this time generating 45 points. The results for the sample image are shown in Fig. 3.9(b). As a final note, the length of the fingernail along the finger direction vector, L_n, is calculated as the distance between the two zero crossing points. This will be used in the next section to find the endpoints of the finger contour.

3.4.2.3 Segmenting the Finger Contour

The final step for adjusting the contours is to segment the finger contour so that the full length of the finger is $1.5L_n$. As with the nail, the most distal point of the finger is found by transforming the finger points into a coordinate system aligned with the finger direction vector and with their centroid at the origin. (This point is marked with a blue circle in Fig. 3.10(a).)

To find the endpoints of the new finger contour, the same algorithm used to find the most-distal point is adapted. First, the rotation angle used in the transformation is $\theta_e = \theta_f + \frac{\pi}{2}$. This rotation means that the algorithm will now search for intersections perpendicular to

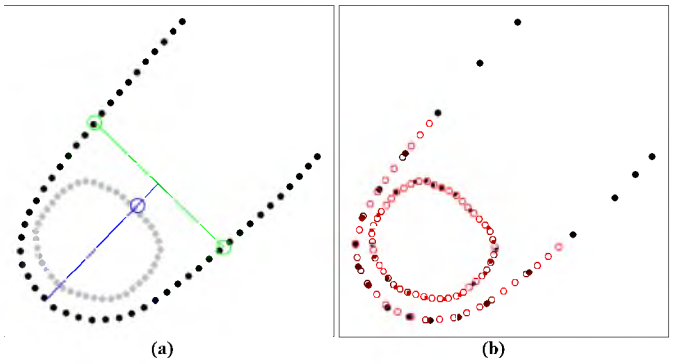

Fig. 3.10. Segmenting the finger contour/final smoothed contour. (a) Segmenting the finger contour using the length of the nail as a guide. The centroid of the finger contour points is the blue circle. The blue line is parallel to $\hat{\mathbf{v}}_f$ and the green line is perpendicular at a distance $1.5L_n$ from the most distal point. The green circles are the new endpoints of the finger contour. (b) Final sample finger contour. Once all segmenting is finished, the remaining contour is smoothed as before. The original points are shown as black dots, while the final points are shown as red circles.

the direction vector. Also, rather than translating the points so their *centroid* is at the origin, a new point must be found that lies along the line that will define the end of the finger. The coordinates of the point P_e locating the end of the finger contour along the finger direction vector is found using the formula

$$P_e = P_0 - 1.5L_n\hat{\mathbf{v}}_f \tag{3.2}$$

where P_0 is the coordinate vector of the most distal point. (The line through the most distal point along $\hat{\mathbf{v}}_f$ is drawn in blue in Fig. 3.10(a), while the line perpendicular to the finger direction vector, which defines the end of the finger, is drawn in green. P_e lies at their intersection.) After transforming the finger contour points to the coordinate system defined by θ_e and P_e, the algorithm now yields two points, one on either side of the finger (marked in Fig. 3.10(a) with green circles). These are the new endpoints of the finger contour. The points between these two (on the distal side) are retained, while the proximal points are discarded.

Once the endpoints have been identified, the finger contour is smoothed again (using 30 points) to provide even spacing along the contour between points, as shown in Fig. 3.10(b). In the figure, the original (hand-selected) contour is shown with black dots, while the final contour is shown with red circles. This procedure ensures consistent generation of contour points across all subjects, potentially allowing for future interfinger registration and force prediction.

3.4.3 Shape Model

The procedure of forming the Shape Model (as well as the rest of the AAM) essentially follows that given in the original work by Cootes et al. [39] and expanded on by Stegmann [75]. However, some details as applied to fingernail images are of interest here. The point coordinates (x_{j_i}, y_{j_i}) for the ith training image are extracted to a column vector of the form $\mathbf{x}_i = \begin{bmatrix} y_{1_i} & \cdots & y_{n_i} & x_{1_i} & \cdots & x_{n_i} \end{bmatrix}^T$.

The shape vectors are then aligned to the mean shape $\bar{\mathbf{x}}$ using Procrustes analysis [76, 77] within an iterative procedure. After alignment is complete, a Principal Component Analysis (PCA) is performed on these vectors to determine the principal modes of variation. The PCA reveals the principal axes, or eigenvectors, $\mathbf{\Phi}_s = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_{2n} \end{bmatrix}$ of the covariance matrix

$$\mathbf{\Sigma}_s \mathbf{\Phi}_s = \mathbf{\Phi}_s \mathbf{\Lambda}_s \tag{3.3}$$

where $\mathbf{\Sigma}_s$ is the covariance matrix of Shapes, and $\mathbf{\Lambda}_s$ is a diagonal matrix of the eigenvalues $(\lambda_1, \cdots, \lambda_{2n})$ corresponding to each of the eigenvectors. The variation in shape may be modeled using the formula

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi}_s \mathbf{b}_s \tag{3.4}$$

where $\mathbf{b}_s$ is a vector of Shape parameters, and $\mathbf{x}$ is an instance of shape. Given (3.4), new shapes may be synthesized.

It is instructive to investigate variation in a single Shape mode. Fig. 3.11 shows the first two modes of variation from the index finger of one subject. The patterns seen here are typical of all subjects. The first mode is the left-to-right motion of the nail across the finger characteristic of x-direction force. The second mode is the up-and-down motion of the nail across the finger characteristic of y-direction force. For all of the PCA models in this work, the number of eigenvectors required to describe 99% of the variation are retained and the rest are discarded.

Another test of the Shape Model is the relationship between the Shape Parameters. If the parameters are dependent, it may signify that the PCA was not a good approximation of the data since it depends on the assumption that the variation in the data is linear. Fig. 3.12 shows plots of pairs of the first three Shape Parameters for the same subject. Since these plots (and those of the rest of the pairs of the Shape Parameters) show no evidence of a relationship between the parameters, it may be concluded that the linear relationship assumed by the PCA holds, and the Shape Model is an accurate representation of the variation in the finger.

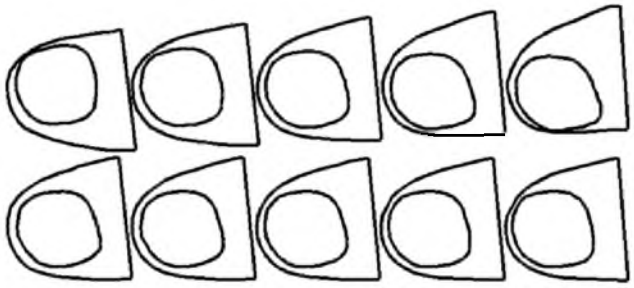

Fig. 3.11. First two modes of Shape variation (subject 10 thumb). The first mode (first row) shows the lateral movement of the nail relative to the finger edge typical of x-direction force. The second mode (second row) shows the longitudinal movement of the nail relative to the finger edge typical of y-direction force.

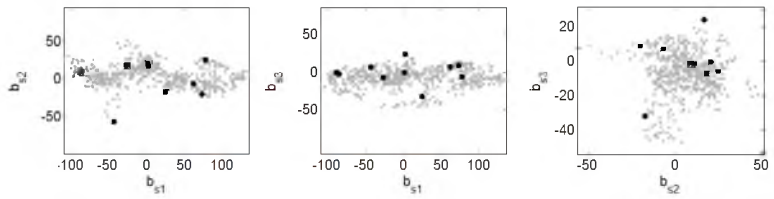

Fig. 3.12. Shape parameter pairs (subject 4 index finger). Gray dots are all images, while black dots represent the training images. Only the first three shape parameters are shown. The lack of relationships apparent in these plots indicates that the linear relationship assumed by the PCA is accurate.

3.4.4 Texture Model

As with the Shape Model, the general formation of the Texture Model adapts the method of Cootes [39]. First, the Training Images are warped to the mean shape using a modified Piecewise Affine transformation. This transformation involves dividing the region of the finger into triangles, then finding the affine transformation T_i that relates the ith triangle in the current Shape to the same triangle in the reference Shape. The Delaunay triangulation is used to determine the triplet of points that form each triangle. The ith transformation is estimated (using least squares) to relate the three vertices in the current Shape to the reference Shape.

To reduce the number of calculations that must be made, only the triangulation of the mean Shape $\bar{\mathbf{x}}$ is calculated for each subject. Although for most subjects the general Delaunay triangulation is sufficient, about 40% (13 of 32) require modifications. This occurs due to two different problems in the contours. First, some contours do not form a convex shape, occasionally containing some points that are within the convex hull. In these cases, the triangles that close the hull but are outside the finger contour are discarded. Second, in

a few cases, the finger shape does not completely enclose the nail. This most often manifests when the nail extends beyond the proximal end of the finger contour, as in the 5th image in the top row of Fig. 3.6. In this case, the nail points that lie outside the finger shape are added to the outside shape, and the triangulation is recalculated.

This mean triangulation is used when transforming all images for this subject/finger combination. For a specific image, the current Shape is transformed to the mean Shape using this standard triangulation. Any triangles that are reversed (where the points have moved too far relative to each other) are discarded.

Once the image is warped, the pixel intensities are raster scanned to a column vector $\mathbf{g}_i = \begin{bmatrix} p_{1_i} & \cdots & p_{m_i} \end{bmatrix}^T$. These Training Image intensity vectors are normalized to reduce global lighting effects. A PCA is then used to determine the principal modes of Texture variation. The details of the PCA are much the same as for the Shape Model:

$$\mathbf{\Sigma}_{g'} \mathbf{\Phi}_{g'} = \mathbf{\Phi}_{g'} \mathbf{\Lambda}_g \tag{3.5}$$

In the texture case, a modified covariance matrix [78] of the form $\mathbf{\Sigma}_{g'} = \mathbf{G}^T\mathbf{G}$, where $\mathbf{G} = \frac{1}{\sqrt{N}} \begin{bmatrix} \mathbf{g}_1 - \bar{\mathbf{g}} & \mathbf{g}_2 - \bar{\mathbf{g}} & \cdots & \mathbf{g}_n - \bar{\mathbf{g}} \end{bmatrix}$. The eigenvectors $\mathbf{\Phi}_{g'}$ of this matrix are related to the eigenvectors of the original covariance matrix using the formula $\mathbf{\Phi}_g = \mathbf{G}^T \mathbf{\Phi}_{g'}$. The variation in Texture may be modeled using the formula

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{\Phi}_g \mathbf{b}_g \tag{3.6}$$

where $\mathbf{b}_g$ is the vector of Texture parameters, and $\mathbf{g}$ is an instance of texture. As before, new textures may be synthesized using (3.6). The first two modes of variation in Texture for one finger are shown in Fig. 3.13. The first mode exhibits the asymmetric pattern formed when x-direction force is applied. The second mode exhibits the alternating bands across the middle of the nail and the distal end of the nail that form under y-direction force. These trends hold across all subject/finger combinations.

As with the Shape model, the relationship between the Texture Parameters is investigated. Fig. 3.14 shows plots of pairs of the first three Texture Parameters for the same subject. As with the Shape Model, there is no evidence of a relationship between the parameters, indicating that the linear relationship assumed by the PCA holds and the Texture Model is an accurate representation of the color variation inherent in the finger.

3.4.5 Appearance Model

The general method is again followed to form the Appearance Model. First, weights are calculated to relate the Shape parameters to the Texture Parameters. Then, the two sets

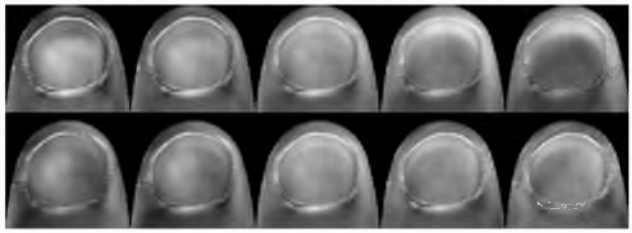

Fig. 3.13. First two modes of texture variation (subject 2 index finger). The first mode (first row) shows the asymmetric patterns characteristic of x-direction force. The second mode (second row) shows the alternating bands across the distal end and middle of the nail characteristic of y-direction force.

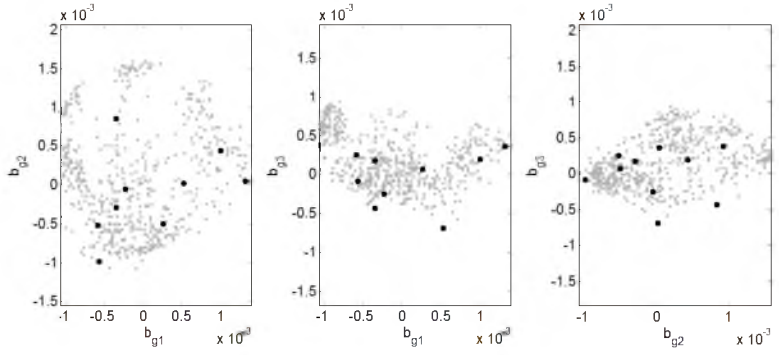

Fig. 3.14. Texture parameter pairs (subject 4 index finger). Gray dots are all images, while black dots represent the training images. Only the first three texture parameters are shown.

of parameters are combined into a new set of column vectors $\mathbf{c}_i = \begin{bmatrix} r\mathbf{b}_{si}^T & \mathbf{b}_{gi}^T \end{bmatrix}^T$. Various methods have been proposed to find the weight values relating Texture to Shape [39, 75]. This work uses the variance ratio $r = \lambda_g/\lambda_s$ of the overall variance in Shape ($\lambda_s = \sum \lambda_{s_i}$) and the overall variance in Texture ($\lambda_g = \sum \lambda_{g_i}$).

Given the combined Appearance vectors, the PCA follows the standard procedure, resulting in the following formula for the variation in appearance:

$$\mathbf{b} = \mathbf{\Phi}_c \mathbf{c} \tag{3.7}$$

where $\mathbf{c}$ is the vector of Appearance parameters, $\mathbf{b}$ is an instance of Appearance, and $\mathbf{\Phi}_c$ is the matrix of eigenvectors. As before, new instances of appearance may be synthesized using (3.7). The first two modes of variation in Appearance for one subject are shown in Fig. 3.15. The first mode exhibits the characteristics of force in the x-direction, while the second mode displays the characteristics of y-direction force.

Again, the relationship between the Appearance Parameters is investigated. Fig. 3.16 shows plots of pairs of the first three Appearance Parameters for the same subject. Since

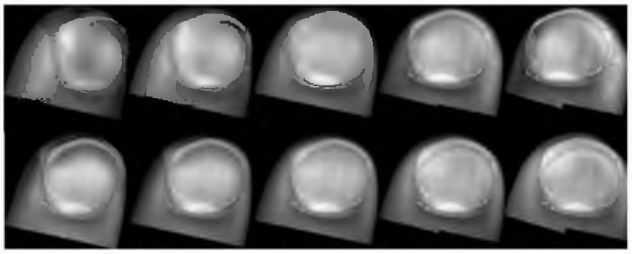

Fig. 3.15. First two modes of appearance variation (subject 4 index finger). The first mode (first row) shows the lateral movement of the nail and corresponding color pattern typical of x-direction force. The second mode (second row) shows the longitudinal movement of the nail and corresponding color pattern typical of y-direction force.

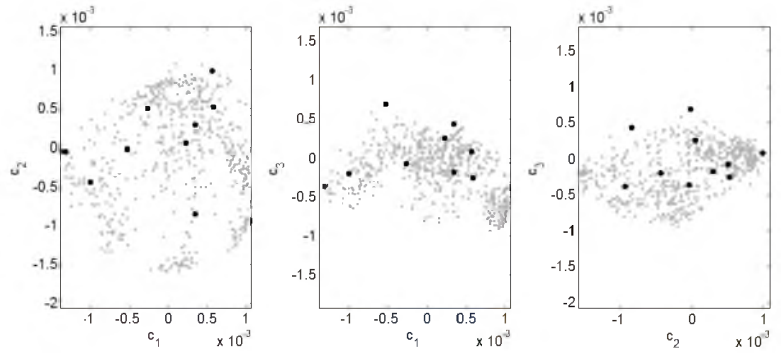

Fig. 3.16. Appearance parameter pairs (subject 4 index finger). Gray dots are all images, while black dots represent the training images. Only the first three appearance parameters are shown.

these parameter-pair plots show no evidence of a relationship between the parameters, it may be concluded that the Appearance Model accurately represents the variation inherent in the data.

3.4.6 Search Model

A four-level, multiresolution Search Model is formed using the standard method of perturbing the training image parameters, resampling the images using the new parameters and calculating the matrix relating the perturbed parameters to the resampled image values [39]. The Search Model is applied to each image in the data set to register each of them. A manual pose estimate with the mean Shape is placed on the first image. The standard AAM Search procedure is then applied, with a maximum of seven iterations for each level of resolution. The search converges when the residual reaches the predefined minimum. Once the search procedure has converged, the point locations are saved. These locations are used as the initial estimate of the pose for the next image. This procedure is

summarized in Fig. 3.2.

For some images, the previous image location does not provide a sufficiently close estimate for the AAM to register the image. In these cases, manual intervention is required, and the investigator stops the registration process to re-align the mean Shape. This generally happens when data collection was interrupted and the finger was repositioned or when the test subject relaxed enough that the controller "surprised" them with a new level of force, causing the finger to apparently jump between one image and the next.

While registering the images, it sometimes becomes apparent that certain images cannot be characterized by the Search Model, regardless of the initial placement of the contour. This is assumed to be an indication that a mode of variation of the data set is not present in the Training Images. Therefore, once all of the images are processed by the Search Model, these images that cannot be processed are grouped. One of the images from this group is randomly selected and added to the set of Training Images. The point contours are manually selected for this new image, and the smoothed finger and nail contours are generated as before. The Shape, Texture, Appearance, and Search Models are all recreated, and the new Search Model is applied to the images in the unprocessed group. This procedure is repeated until no unprocessed images remain. Most data sets are successfully registered with no additional Training Images; however, 6 of the 32 data sets required one extra Training Image. No sets required more.

3.5 Registration Error

One may reasonably inquire as to whether the images have been registered correctly when the process is complete. The general procedure for determining correct image registration is to subtract pixel intensities from one another. However, since the intensity at a given location in these images changes from one force level to the next, repeated images would need to be recorded at each level to verify registration using this method. Since time constraints prohibit such a large data set from being collected, the current image is instead compared against other "nearby" images, where the Cartesian distance between images in Force Space is used. (It is assumed that, when the finger is experiencing similar forces, the Shape, coloration, and other properties should also be similar.) This distance metric d is

$$d_{ij}^2 = (f_{xi} - f_{xj})^2 + (f_{yi} - f_{yj})^2 + (f_{zi} - f_{zj})^2 \tag{3.8}$$

A thresholded Gaussian kernel method is used to determine the weight each other image should carry for any given image, similar to that used in Locally-Weighted Polynomial Regression [56]. The weight for the jth image (relative to the ith image) is

$$w_{ij} = \begin{cases} e^{-\frac{1}{2}\left(\frac{d_{ij}}{K}\right)^2} & d_{ij} \leq d_c \\ 0 & d_{ij} > d_c \end{cases} \tag{3.9}$$

for the kernel width K and cutoff distance d_c. The weighted average of all nearby images is calculated, and this average is used for comparison. For the data analyzed in this paper, the kernel width is set to $K = 0.25$, and the cutoff distance is set to $d_c = 2K = 0.5$. Using these parameters, at least 2 nearby images are used to form each average image, while many combine over 100 nearby images. The RMS error in pixel intensity between the current image and the average image is then found. Outliers are investigated as potentially poorly registered images and the registration corrected, if needed. For each subject/finger combination, the overall RMS error is also found. No statistically significant difference is found between the effect on the registration accuracy in the thumb images (1.5 ± 0.3) or the index finger images (1.4 ± 0.3).

3.6 Method Modifications

The standard method presented thus far has been modified, in an attempt to characterize its behavior and determine optimal parameters. The three alterations that are considered here are (1) changes to the number of landmark points used to represent each contour, (2) selection of the color channel to be processed, and (3) shape of the underlying model. Each of these modifications is detailed, along with the corresponding results, in the following sections.

3.6.1 Landmark Point Density

The first parameter that has been varied in these experiments is the density of the landmark points in the contours used for AAM registration. A preliminary investigation into the variation of force calibration accuracy with point density was conducted previously [37]. The motive is to reduce the computational complexity. The Standard model uses 75 landmark points, with 45 points on the edge of the fingernail and 30 on the outside edge of the finger. When registering images between subjects such density may be necessary as different shapes of finger and nail require more complete models to ensure accurate mapping. When registering images that all belong to the same finger, however, the nail remains rigid, with the only apparent deformation occurring due to rotations out of the viewing plane. A less complex model should suffice and reduce the number of calculations required to form and apply the AAM.

Various landmark point configurations were designed in order to characterize the rela-

tionship between landmark point density and registration error. The Standard model is used as the basis for the other models. For consistency, the same training images are used. The number of landmark points is reduced using the same interpolation procedure explained previously. Fig. 3.17 shows the five different landmark point configurations (containing 75, 50, 25, 15, and 6 points) applied to the same image. This process is repeated for each of the training images in the original data set. The AAM is formed as before and the rest of the image set is registered.

The registration error for these images is calculated as follows. First, each of the registered contours from the Standard model is interpolated as if it were one of the Training Images. Then the corresponding image is transformed to the mean image for the current reduced-point model. The pixel intensity RMS error for each image is calculated as before. These errors are aggregated for each subject into a single RMS value. The overall mean and 95% confidence intervals for each Point Density registration model are shown in Fig. 3.18. Since the error for the Standard model is calculated using a different method than the others, it should not be considered an adequate benchmark for the other error values. It is more appropriate to compare the four reduced-count models. The mean Pixel Intensity RMS errors for the respective models are 3.3 ± 0.7 (50-point model), 4.0 ± 0.6 (25-point model), 5.1 ± 0.8 (15-point model), and 6.8 ± 1.4 (6-point model). Using ANOVA followed by Tukey's HSD test, no significant difference is detected between the 50-point model and the 25-point model, while the lower density models do show a statistically significant difference. Thus, a conclusion may be drawn that the point density can be reduced to the 25-point model without significantly affecting the registration results. It should be noted, however, that even the largest error represents approximately 2.7 % of the full intensity range of 0–255.

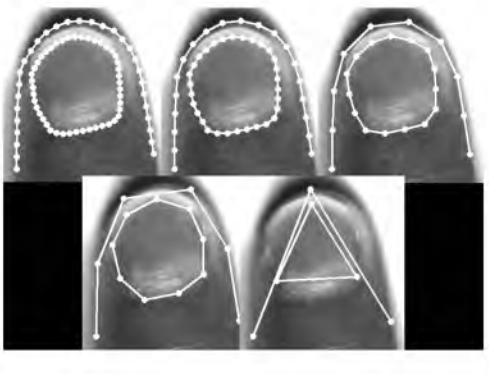

Fig. 3.17. Different landmark point configurations for the same image, with (a) 75, (b) 50, (c) 25, (d) 15, and (e) 6 points.

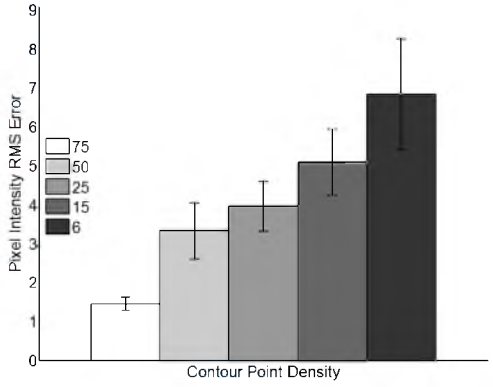

Fig. 3.18. Pixel intensity RMS error of the point density registration models (scale 0–255). The 75-point model is used as the standard for the other models, so only comparisons among the reduced-point models are valid. The 15- and 6-point models have significantly higher error than the 50-point model.

3.6.2 Color Channel Registration Models

The color channel used during registration was the second parameter to be investigated. A preliminary investigation into the variation of force calibration accuracy with color channel was conducted previously [37]. The Standard model uses the full RGB images to form the AAM. However, since the finger coloration effect is most pronounced when reflecting green light [4, 64], the force prediction model uses only the Green channel to predict force. If the Green channel alone could be used to register the images, processing time would decrease, since the full RGB images require three times as many calculations. If it is demonstrated that the registration accuracy is not reduced when using a Green-only AAM for registration, this change may be freely implemented.

Three other color-based registration models are also implemented: Gray (in which the grayscale average of the three color channels is taken before model formation or image registration), Red-only, and Blue-only. It is expected that the registration algorithm will work nearly as well for these models since the color information contained within all of the channels is roughly the same. However, force prediction should be less effective using the Red-only AAM since red light has been shown in the past to be less effective when reading the force coloration effect [64]. To form the color-specific AAM, the same 9 or 10 training images and landmark point locations are used across all five models. As with the Landmark Point Density models, this ensures a consistent starting point.

Estimating the error for these registration models is straightforward: since the number of points remains the same, no interpolation is needed. The images are registered to the new template (e.g., the Red model) using the points from the Standard model and the

triangulation from the new model. The Pixel Intensity RMS error is again calculated, and the error in all images for one subject is calculated as a representative value for that individual. The mean and confidence intervals for the individual errors are shown in Fig. 3.19 for each of the color channel registration models. As with the previous error comparison, the error from the Standard model is included only for reference. The Gray (4.8 ± 0.9), Red (4.9 ± 0.6), Green (5.5 ± 0.9), and Blue (6.4 ± 1.1) models exhibited mean errors with no statistically significant difference. As before, all numbers are given in the unitless pixel range of 0–255.

3.6.3 Modified Shape Registration Model

The final method of modifying the standard AAM is to extract only the skin along the sides of the nail, rather than the entire skin shape. The motivation behind this alteration is that the skin along the distal end of the finger tends to deform much more than the skin along the sides of the nail while not providing as much useful color information for force prediction. When it deforms, the apparent space between the nail and the edge of the finger changes, causing features in this region (such as the free matrix, hangnails, etc.) to occupy a larger or smaller percentage of the region. When the image is warped to the standard image, the nail appears to change shape in response to force. Obviously, this effect is undesireable. Eliminating the distal region of the skin from a force prediction model would be ideal since it would remove this dependence on deformation-based pixel intensity changes. However, it would be useful to know whether removing these regions negatively impacts registration of the images.

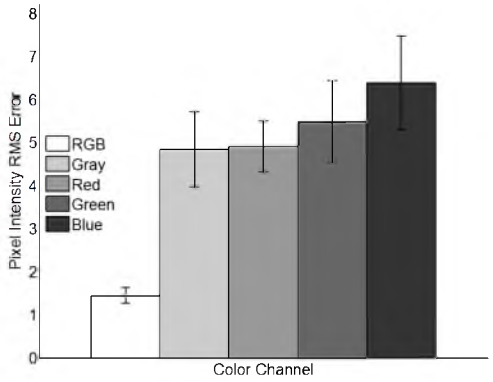

Fig. 3.19. RMS error of the various color channel registration models (scale 0–255). The RGB model is used as the standard for the other models, so only comparisons among the alternate color-channel models are valid. No statistically significant difference among the errors is detected.

This method is illustrated in Fig. 3.20(a). The nail contour remains identical to that used in the standard AAM. However, the finger contour is segmented using the following algorithm: The nail centroid, the finger direction vector, and the most proximal point of the nail were all found in Section 3.4.2.2. A line is drawn perpendicular to the finger direction vector, through the nail centroid. The intersection of this line with the finger contour is found using the method of that same section. The same procedure is used to find the intersection of the finger contour with the line perpendicular to the finger direction vector through the most proximal point of the nail. These two pairs of points are used to divide the finger contour into segments along either side of the nail, which are then smoothed as described previously. For this work, only 5 points are used to smooth along each side of the finger since the edge of the finger is relatively straight in this region, as illustrated in Fig. 3.20(b).

The general Delaunay triangulation would include much of the skin along the upper sides of the nail that is not desired in the Texture formulation. Therefore, the modified algorithm detailed previously is applied in three parts. The nail contour is used as the first region, and each of the two sides are extracted as separate regions. The side regions are bounded by the proximal end of the nail on one side and by the nearest nail contour point on the distal side. Only the triangles inside each region are retained. A typical triangulation is shown in Fig. 3.21.

From here, the procedure is the same as that used to form the Point Density models. The Training Image shapes for a given subject are converted to the Sides model using

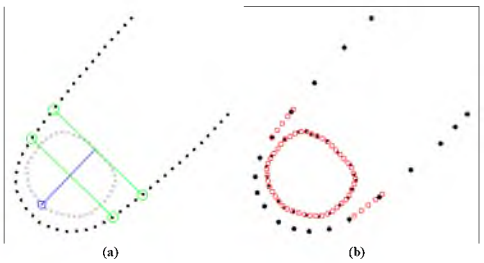

Fig. 3.20. Segmenting the finger contour to use only the skin along the sides of the nail. (a) The nail contour is adjusted as before. To segment the finger contour, the line perpendicular to the finger vector is drawn through the nail centroid (lower green line). This marks the most distal point of the finger contour on both sides (lower-left green circles). The most proximal points of the finger contour (upper-right green circles) are found by drawing the line perpendicular to the finger vector through the point along the nail contour that intersects the finger vector line in the proximal direction (upper green line). (b) The final smoothed contour is superimposed over the original points.

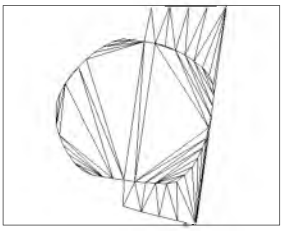

Fig. 3.21. Modified Delaunay triangulation of the modified shape registration model. The procedure begins with the standard Delaunay triangulation. Lines are drawn between the ends of the finger contour, and triangles outside this region are eliminated.

the above procedure. Then, a new AAM is formed using these newly annotated Training Images. The new AAM is used to register the other images in the data set. To calculate the registration error, the point contours found using the Standard registration model are converted to the Sides model using the above procedure. The Pixel Intensity RMS error is then calculated for all images and the RMS error for each subject is found. The mean RMS error for the Register Sides model is 4.7 ± 0.6.

3.7 Force Prediction

The final goal, of course, is not merely to register the images, but to calibrate a force prediction model using these registered images. The EigenNail Magnitude model, whose derivation has been detailed previously [36], is used to evaluate the force prediction capabilities of each of these registration models. In essence, this model uses Principal Components Regression [79] to relate the forces to the pixel values, thus reducing the unwieldy M-by-N (images by pixels) input matrix to a more manageable M-by-P (images by EigenNail coordinates) matrix. The first P EigenNails are found that represent 99% of the variation in the pixel data, and all of the images are projected onto these P EigenNails to form coordinates in Nail Space. A least-squares regression is performed to relate these Nail coordinates to the forces in x, y, and z simultaneously.

To validate the models, each subject's registered images are randomly divided into Training and Validation sets, with 25 % of the images selected as Validation data. The model constants are calculated using the Training data. Then, the model is used to predict force on the Validation data. This simulates the situation where a representative calibration set has been taken and a new data set is also taken, but may not contain exactly the same forces as the calibration set. One hundred such validation experiments are simulated for each subject, with a different validation set chosen each time. The aggregate prediction RMS error is calculated for each subject. In Fig. 3.22, the Standard model results are

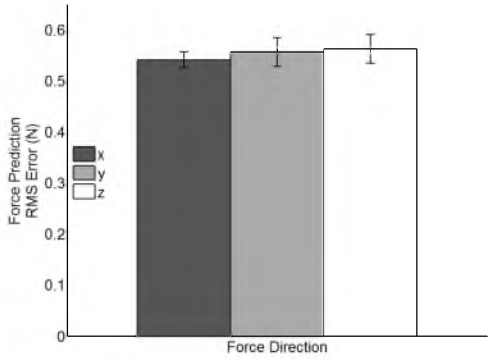

Fig. 3.22. Force Prediction RMS Error of the Standard registration model, grouped by force direction. No statistically significant difference is detected among the three force directions.

plotted, showing the mean RMS error in each force direction, along with the 95 % confidence intervals. All three directions of force exhibit RMS errors of 0.56 ± 0.03 N or less.

This same method is used to determine whether any significant effect may be seen among the three model variations. Fig. 3.23 shows the comparison among the five Landmark Point Densities. The standard 75-point model exhibits an overall RMS error of 0.55 ± 0.01 N, while the 50-point, 25-point, and 15-point models each exhibit 0.56 ± 0.01 N of RMS error. The 6-point model shows a significantly higher error of 0.62 ± 0.02 N.

The Color Channel registration model comparison is shown in Fig. 3.24. As before, the Standard (or RGB) model has an RMS error of 0.55 ± 0.01 N, while the Gray model has an error of 0.57 ± 0.01 N. The Red model has an error of 0.57 ± 0.02 N, while the Green and Blue models have an error of 0.56 ± 0.01 N. No significant difference may be seen between any of the models. This result is surprising, given that past results [57] have shown that the green channel correlates more closely with force prediction than other channels. However,

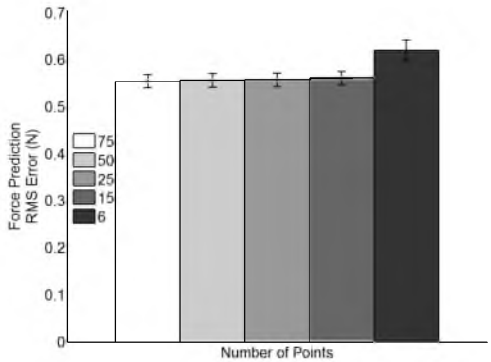

Fig. 3.23. Force prediction RMS error of the landmark point density registration models. The 6-point model has a significantly higher error than the rest.

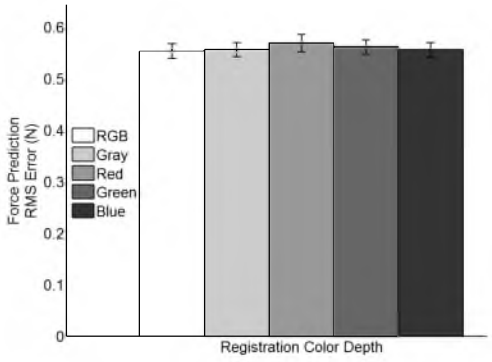

Fig. 3.24. Force prediction RMS error of the color channel registration models. No statistically significant difference in the error is detected.

it may be due to variations in lighting as previous data collection used an unspecified "soft" white LED source, while the current data set used super-bright white LEDs (Model No. YSL-R547W2C-A13). In addition, prior experiments did not control the lighting as closely, allowing ambient light to reflect from the fingernail into the camera.

The comparison of the registration model Shape is shown in Fig. 3.25. An additional model is shown here since the comparison may be made between the Standard model, the "Register Sides" model (which was discussed previously), and the "Convert Sides" model, in which the force is predicted from the images registered using the Standard model but then converted to the short Sides-only model. Again, beginning with the Standard model's 0.55 ± 0.01 N mean RMS error, the Register Sides and Convert Sides models both exhibit a slightly higher (albeit statistically significant) error of approximately 0.60 ± 0.02 N. Note all of these results in light of the resolution of the force encoding method (0.047 N).

In addition, the time required to predict force on 100 randomly selected images from each

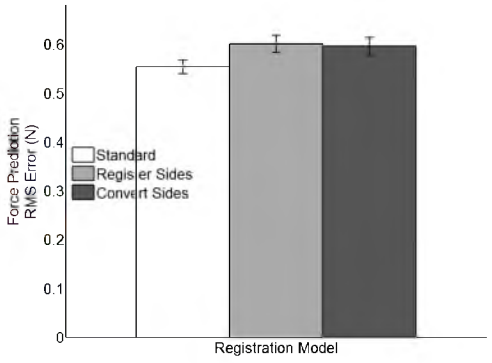

Fig. 3.25. Force prediction RMS error of the modified shape registration models. The Standard model has a significantly lower error than the other two models.

subject is calculated. In these experiments, the Standard (RGB) 75-point registration model predicts force at an average rate of 0.63 ± 0.01 ms per image. The 50-point and 25-point models exhibit statistically similar times (0.62 ± 0.01 ms and 0.62 ± 0.01 ms, respectively), while the 15-point model (0.58 ± 0.01 ms) and the 6-point model (0.43 ± 0.01 ms) are statistically different. Likewise, the Gray (0.67 ± 0.01 ms) and Green (0.66 ± 0.01 ms) models predict force significantly slower than the RGB model but similar to each other, while the Red (1.04 ± 0.02 ms) and Blue (0.94 ± 0.02 ms) are significantly slower than the other three. Likewise, the Convert Sides (0.39 ± 0.01 ms) and Register Sides (0.37 ± 0.01 ms) models predict force significantly faster than the Standard model.

3.8 Conclusion

A new registration method has been demonstrated and developed that registers fingernail images with an RMS pixel intensity error of approximately 1.5 ± 0.3, an error of less than 0.5% of the total intensity range of 0–255. Three-dimensional force prediction based on these images has an RMS error of approximately 0.55 N, which represents 9.2 % of the full range of forces measured in the experiments. The marked improvements of this method over prior registration methods are (1) no need for fine-tuning of registration method parameters for each individual, (2) no images are discarded because they cannot be registered, (3) a registration model based on actual variation of the finger, rather than general image processing techniques, and (4) the force prediction model is no longer training to the residual motions in the registered images.

In addition, the new registration method has been refined according to the needs of fingernail imaging. Specifically, the following improvements to the Standard model have been made: (1) the number of landmark points may be reduced to as low as 25 and (2) the green color channel may be used to form the registration model rather than using the entire RGB image or the grayscale average of the color channels. Both of these have been shown to result in no significant loss of accuracy in registration or force prediction.

In spite of these advances, some questions have arisen due to the investigation as well as some avenues for future inquiry. First, a simple improvement to the technique used to encode forces from the calibration used here may shed more light on the significance and accuracy of the results. Second, the registration method still requires some human intervention. It may be possible to more fully automate the process by implementing an edge-detection procedure to position the initial estimate of the contour prior to the search as well as for the Training Image contour Landmark Point placement. Additionally, it may

be possible to further automate the error checking when the contour estimate is unable to converge to a good solution, using techniques similar to the registration error calculation for the Standard model.

Finally, it is the goal of the authors to show that this registration method and the associated force prediction models may be used to successfully perform a human grasping experiment, wherein the individual finger forces are measured only using the coloration of the fingers, without mediating force sensors. This would allow test subjects to freely choose both the contact locations and modality of their grip, improving researchers' ability to measure natural grasps.

CHAPTER 4

OPTIMIZING FINGERNAIL IMAGING CALIBRATION FOR FORCE PREDICTION

4.1 Abstract

This paper discusses the optimization of a fingernail imaging system for predicting fingerpad force. Three aspects of fingernail imaging are investigated: the lighting coloration, the calibration grid design, and the models used for force prediction. White and green LEDs are found to produce statistically similar effects on registration error and force prediction results across all three directions of force. Two calibration grids are implemented with no statistically significant difference in either registration or force prediction between the Cartesian and cylindrical grid designs. Of the five force prediction models investigated, a principal component regression model based on the pixel intensity eigenvectors estimates the force with the greatest accuracy.

This EigenNail Magnitude Model simultaneously estimates force in all three directions with RMS error of $0.55 \pm 0.02\,\mathrm{N}$ (9.1 % of the normal force range of 6 N). The other four models exhibit significantly larger RMS errors, from $0.65 \pm 0.03\,\mathrm{N}$ to $1.30 \pm 0.09\,\mathrm{N}$. The calibration grids result in force prediction errors less than $0.57 \pm 0.05\,\mathrm{N}$ in all directions. White LEDs correlate with a force prediction RMS error of $0.55 \pm 0.01\,\mathrm{N}$, while green LEDs correlate with an error of $0.59 \pm 0.02\,\mathrm{N}$. These results indicate a set of optimal parameter choices for the calibration of a fingernail imaging system.

4.2 Introduction

Finger force estimation using fingernail images is a method with some history and multiple attempts at refinement [1, 3, 27, 59]. Fingerpad force is transduced using the naturally occurring color patterns present in the fingernail and surrounding skin. When the tissue beneath these surfaces is deformed by the incident force, the blood flow through those tissues is impeded, creating patterns that correlate to the magnitude and direction of force on the finger. These patterns have been observed across the entire population [2].

The original fingernail sensors, consisting of a series of photodetectors and infrared LEDs in a sheath that was custom-manufactured for each user, were capable of simultaneously measuring three-dimensional normal forces up to 3 N and shear forces up to ±2 N, with an RMS error of 1 N normal force and 0.5 N shear force [59]. Multiple improvements have been made to this method. Stretchable materials have been incorporated into the design [6], removing the need for individually fitted sensors. In addition, the effects of the optical path length (the distance between the LED and the photodetector) and the wavelength of light emitted by the LEDs have both been optimized [5]. Green light has been found to correlate with the optimal response rate of the sensors.

Fingernail imaging is an additional technique based on the same principle, using digital cameras and controlled lighting to image the finger [3]. Using this method expands the imaging capability to include a larger area of the finger while increasing the resolution. The errors in force sensing are likewise improved, resulting in RMS errors of 0.3 N in normal forces up to 6 N and 0.5 N in shear forces up to 3 N. The modeling complexity required to achieve this accuracy was reduced in that the model was trained on only one or two components of force at once, rather than simultaneous three-dimensional force prediction. Although the authors are not aware of any formal investigation into the effects of green light on the force prediction accuracy of fingernail imaging, past results have shown that the green channel of an RGB image does correlate most closely with the force [57]. The first parameter investigated herein is the light color. No significant difference is found between white and green light in either registration error or force prediction error.

Previous work has relied on the test subject to exert forces on a stationary sensor during data collection for calibrating the force prediction models. An automated calibration routine has been developed that permits the subject to be more passive during calibration [67], which allows a larger data set to be collected. Where 300 images may have been the limit during a manual calibration experiment, nearly 3000 images may be collected in a reasonable time using automated calibration. With the increase in collection ability, the concern becomes how best to arrange the target calibration set within the desired force space to maximize the observability [80] and identifiability [70] of the prediction model parameters. Therefore, the effect of calibration grid choice on force prediction error is also characterized in this paper. No significant difference is found between the two grid types in either registration or force prediction error.

Several models relating finger force to pixel intensity have been proposed. A generalized least squares technique uses a linear model to relate each force direction to all of the

pixel intensities in the finger [3]. Several forms of Principal Component Regression have been used to relate the forces to either the pixel intensities or the Active Appearance Model parameters [36]. However, the relative merits of these various models have not been investigated. Thus, the third and final purpose of this paper is to compare the existing models and determine which is most successful at predicting force. The EigenNail Magnitude Model, originally introduced in [36], is found to have significantly lower force prediction error than all of the other models examined. In addition, a qualitative analysis method is introduced, which demonstrates that the models are trained to the expected patterns found in the finger.

In preliminary work [36,37], three of the five force prediction models presented here were developed. This work expands on those results in several ways. First, two new models, the Shape Parameters Model and Appearance Parameters Model, are introduced. Second, the effectiveness of all five models is examined across a larger data set, including 64 combinations of subject, finger, and LED color. Third, a standardized validation method is demonstrated, which may be used in future work to determine the effectiveness of new models. Fourth, to demonstrate the effectiveness of the model training, a qualitative analysis method is developed. Fifth, the prior work used only white LEDs and did not examine the effect of calibration grids on the results.

This paper presents the experimental setup, including details of the lighting method and calibration grids, in Section 4.3. The force prediction models are explained in Section 4.4, and the experimental results are presented in Section 4.5.

4.3 Calibration Setup

4.3.1 Experimental Apparatus

To perform the calibration, a Magnetic Levitation Haptic Device (MLHD) [61] is employed. The MLHD has a translational workspace with a 12 mm radius and a rotational workspace of 8° in all three directions. It is capable of exerting up to 20 N of three-dimensional force and 4 N · m of three-dimensional torque.

An ATI Nano17 6-axis Force/Torque sensor is attached to the flotor of the MLHD to measure the contact force. A rubber surface is attached to the sensor to provide comfort, and a 3 cm square of two-sided tape is affixed to improve frictional behavior. The sensor is capable of detecting normal forces up to 17 N and shear forces up to 12 N with a resolution of 0.78 mN. This range and resolution is sufficient for the calibration as the intended ranges are 10 N normal and 5 N shear.

A Point Grey Research FLEA video camera is mounted over the finger to record images during the calibration. This camera records 1024×768-pixel RGB images during calibration. The camera is attached to a static frame since the location of the finger does not vary substantially during calibration.

A lighting box is placed above and behind the finger to provide uniform illumination during calibration. This box consists of an array of 140 LEDs with a piece of tracing paper (3-lb. weight) to diffuse the light and reduce the shadowing effect due to the internal structure of the LEDs. The box is placed on a gooseneck so that it may be adjusted for individual variation prior to each experiment. Positioning the light source above and behind the finger ideally eliminates any glare on the nail, allowing only that light which penetrates into the nail bed to reflect back to the camera. Two lighting boxes are created, one containing white LEDs and the other containing green. A light shield is mounted over the camera to block ambient light.

Past research [59] has found that the metacarpophalangeal (MCP) joint angle has no effect on the coloration of the finger. The distal interphalangeal (DIP) and proximal interphalangeal (PIP) joints, on the other hand, do affect the blood flow to the finger (and hence the coloration). Direct control of either the DIP or PIP joints without affecting blood flow or obscuring the camera's view of the finger has thus far proved impossible. However, they can be controlled in the sense that the angle of the proximal phalanx can be controlled relative to the force sensor. If this angle is set correctly, the finger is positioned to encourage test subjects to maintain a constant angle in each of the DIP and PIP joints. To position the finger during calibration, a restraint device is implemented using a Roylan Static Progressive Finger Flexion Splint. The straps, which would ordinarly be used to hold the splint on the finger, restrict blood flow and so are removed. Rather than restrain the finger's movement, the splint is used to guide the subject in finger placement.

The calibration setup is illustrated in Fig. 4.1. A restraint controls the proximal phalanx angle and positions the finger above the flotor. The camera is placed directly above the finger. The light source is located on a flexible support above and behind the finger. Fig. 4.2 shows a picture of the experimental apparatus.

4.3.2 Calibration Grids

The effect of different types of calibration grids on the calibration process is explored. The calibration grids chosen, as shown in Fig. 4.3, are a Cartesian (x, y, z) grid and a cylindrical (r, θ, z) grid. The grid spacing is selected such that the number of points is approximately the same.

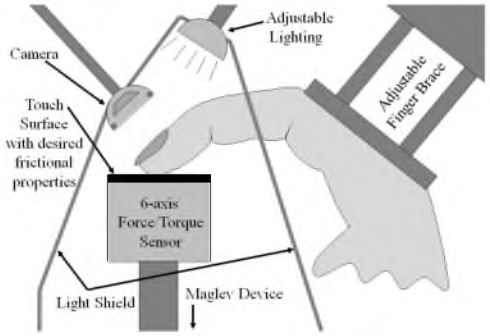

Fig. 4.1. Schematic of calibration setup. The flotor exerts force against the fingerpad with a flat contact surface, while the finger rests against the restraint to guide the approximate angle of the proximal phalanx. The camera is positioned above the nail, with the light source above and behind the finger. The light shield blocks glare from ambient light.

Fig. 4.2. Photo of calibration setup. The camera, mounted on a fixed base, is positioned above the finger, with the light box above and behind the finger. The restraint positions the finger with respect to the flotor, which contacts the fingerpad. To allow the camera to view the scene, the light shield has been partially removed.

The grids are generated with closer spacing at lower levels of normal force to increase sensitivity. At normal forces below 1.5 N, $\Delta F = 0.25$ N, while above 3.0 N, $\Delta F = 1.0$ N. Between these limits, $\Delta F = 0.5$ N. For the Cartesian grid, the forces are spaced evenly in all three directions. In the cylindrical grid, the normal force and the shear force magnitude likewise are distributed evenly, while the angular spacing ($\Delta\theta = 30°$) does not change across the range of normal force. The Cartesian grid covers the force space with 194 points, while the Cylindrical grid uses 250.

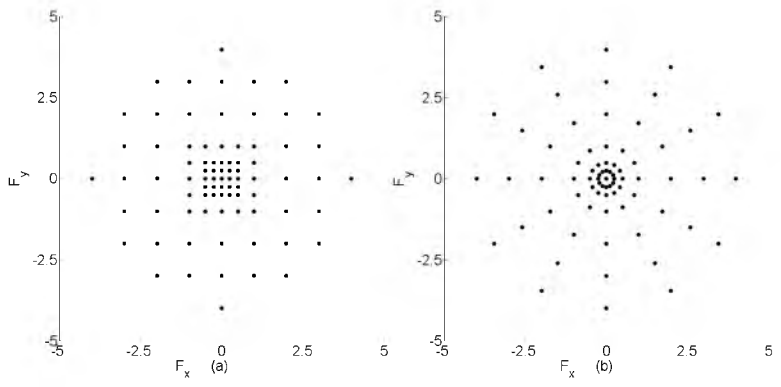

Fig. 4.3. Calibration grid design. (a) Cartesian vs. (b) Cylindrical grids. Grid spacing is chosen so that approximately the same number of points are generated in both grids.

4.3.3 Experiment Design

Data from 16 subjects are collected. Each subject sits for a period of 2 hours while images and forces are recorded on both the index finger and the thumb. Images are recorded only once at each force level. These images and the corresponding forces are used to calibrate the force prediction model. Four data sets are collected from each individual, representing all combinations of LED color (white vs. green) and finger (thumb vs. index). Each subject is given a brief introduction to the task prior to the experiments.

Past experiments showed a training effect when all calibration data followed consistent motions. For example, the on-nail sensors [1] were trained using a calibration trajectory that always rotated counter-clockwise. A bias was detected in the data that may have been due to this rotation. True randomization in the calibration trajectory would eliminate such biases. Since such a randomized trajectory might require the MLHD to transition quickly between very different force conditions, it could cause unstable operation. A different, more structured randomization method is chosen.

To reduce the influence of point order on the results, each calibration grid is divided into nine regions, with a central region consisting of shear forces near zero, and eight wedge-shaped regions containing ranges of shear force angles. For a single Finger/Color combination, a subject experiences all of the forces in each region three times, with the overall region order being randomized for each combination of test subject, finger, and LED color. Within a single region, the target point order always remains the same, to prevent large motions through the workspace (and thus, unstable operation). Thus, four data sets are recorded per subject, each using a randomized test order, containing between 580 and 750 images and the corresponding force readings.

Two-minute rest breaks are provided after every nine trajectories. In addition, after the

finger experiences all 27 trajectories, a longer, 5-minute break takes place while the light box is changed. For some subjects, the two fingers are calibrated on different days. For those that experience the entire 2-hour session at once, a third, 10-minute rest period passes between finger sessions while the experimental platform is modified to accommodate the changes between finger and thumb.

4.3.4 Image Registration

Image registration is accomplished using Active Appearance Models (AAM) [39]. Following a method developed previously [36, 37], each finger's images are used to form an AAM. The Search Model is then applied to register all images pertaining to that particular finger. Following registration, the images are used to form force prediction models.

4.4 Force Prediction Models

Five different force prediction models are used, two of which are significant extensions of previous models, and the other three of which are entirely new. All five predict force simultaneously in three dimensions.

4.4.1 Linearized Sigmoid Model

The Linearized Sigmoid Model is based on the generalized least squares force prediction model used in prior experiments [3]. Previous use of this model, however, only implemented prediction of a single direction of force at a time and trained the model on a data set that only included images from a single direction of force at one time. This model does not readily expand to three dimensions of force, as will be demonstrated shortly. The response of the ith pixel to force in the j-direction (f_j) has a shape such as that shown in Fig. 4.4, approximately following a sigmoid curve with an upper and lower saturation limit and a nearly linear region in the middle. The original model involved fitting sigmoid curves using nonlinear least squares. The data for each pixel was fit to the model

$$p_i(f_j) = a_i + \frac{b_i}{1 + e^{c_i(f_j - f_0)}} \tag{4.1}$$

To reduce computational complexity, several simplifications were made. First, a locally weighted linear regression procedure was used to fit the data rather than nonlinear least squares. Second, the gradient of this fit was estimated, and all points with gradients above 20% of the maximum gradient were marked to lie within the force transducing range. The corresponding pixel intensity value range was called the response range. The saturation limits were estimated by finding the maximum and minimum values of this response range.

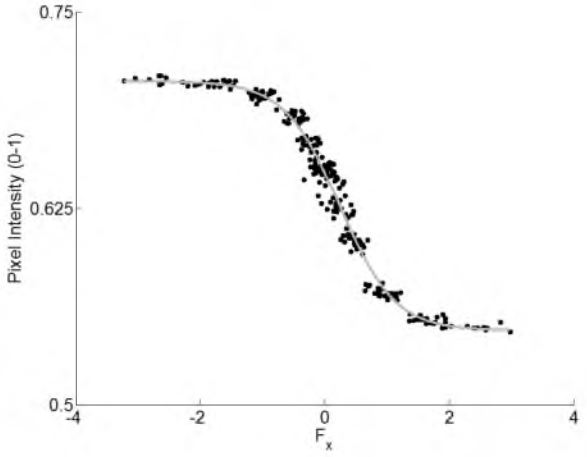

Fig. 4.4. One-dimensional force response of a single pixel. In general, either a pixel does not respond to force in a given direction, or fits the model of (4.1).

If this range was too small (less than 5 units on a scale of 0–255), the pixel was rejected as not varying enough. Third, the data within the range were assumed to fit a straight line. The correlation coefficient of this line was calculated and, if it was less than a certain threshhold (0.8), the pixel was rejected as insufficiently linear. The line models of all pixels that were not rejected were combined to form a single prediction model.

When pixel response data are collected that include full three-dimensional force, the response of any given pixel does not typically follow such a simple curve (Fig. 4.5). The same pixel may respond to force in all three directions, and the force in one direction may

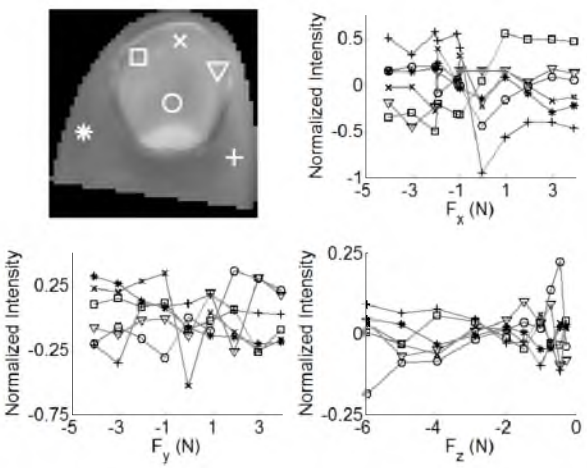

Fig. 4.5. Three-dimensional force response of several pixels. The response of several pixels to F_x, F_y, and F_z are shown. Each pixel is marked with a different symbol. The pixel intensities (in the range 0–1) are centered about the mean value for each pixel to better display trends for all pixels.

affect the response to force in another. Different directions of force respond at different levels of force, if at all. Visualization of the response pattern is difficult at best, given that the overall model must now include four variables (f_x, f_y, f_z, and p_i).

Although a nonlinear least squares procedure could be used to fit this model, it has instead been determined to proceed as before. First, the values for the ith pixel across all images are extracted into a single vector, $\mathbf{p}_i$, and a locally weighted polynomial regression procedure [56] is implemented to find the local four-dimensional hyperplane that fits each point. The maximum gradient of this fitting function is found, and any data points with gradients more than 20 % of this maximum are designated to lie in this response range. (No transducing range is designated as it is found that the three-dimensional shapes are complex and not easily specified.) The saturation limits are used in the same manner: if this range is less than 5 units on the scale of 0–255, the pixel is rejected as not varying enough. As before, the data within this range are assumed to be linear. However, it now must be fit to a four-dimensional hyperplane, with the equation $p_i = a_i f_x + b_i f_y + c_i f_z + d_i$. If the correlation coefficient of this hyperplane is less than the threshold (0.8), the pixel is rejected as insufficiently linear.

Finally, the covariance matrix ($\mathbf{\Sigma}$) is approximated by a diagonal matrix consisting of the variances of each of the M retained pixels. The final assembled model is

$$\mathbf{F} = \left(\mathbf{L}^T \mathbf{\Sigma}^{-1} \mathbf{L}\right)^{-1} \mathbf{L}^T \mathbf{\Sigma}^{-1} \left(\mathbf{p} - \mathbf{d}\right) \tag{4.2}$$

where $\mathbf{L} = \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{bmatrix}$ is the coefficient matrix whose columns consist of the hyperplane coefficients (a_i, b_i, c_i) and $\mathbf{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_M \end{bmatrix}^T$ is the pixel offset vector. Once an image has been registered, (4.2) may be applied to estimate the force.

4.4.2 EigenNail Magnitude Model

The EigenNail Magnitude Model is based on a force direction classifier, which was developed previously [27]. In prior experiments, images were collected at discrete force levels and a Principal Component Analysis (PCA) was performed to determine the eigenvectors (called EigenNails because they reveal the major color variations of the fingernail). Only the first k EigenNails, representing 99 % of the variation in the data, were retained. A Linear Discriminant Analysis was used to determine the weights of each image along the eigenvectors and find a classifier to separate the force directions.

The new model expands on this concept by recognizing that the eigenvectors form a vector space that spans a k-dimensional subspace (hereafter referred to as the "Nail Space") of the entire M-dimensional pixel space. Using this terminology, what were referred to as

"weights" in previous work may now be called "coordinates" in Nail Space. Given that a classifier successfully separated the six groups of force, it is hypothesized that a linear relationship exists between the Nail Space coordinates and the force associated with the image. A linear multivariable least squares regression fit is determined between these Nail Space coordinates and the forces. Once the images have been registered, the pixel values for each image are arranged in column vectors, $\mathbf{g}$. A PCA is performed to calculate the mean vector $\bar{\mathbf{g}}$, the eigenvalues λ_i, and the eigenvectors (EigenNails) $\mathbf{v}_i$. Only the first k EigenNails are retained, to retain 99% of the variation in the data set. The EigenNail model may then be expressed as

$$\mathbf{g} = \bar{\mathbf{g}} + \boldsymbol{\Sigma}\mathbf{w} \tag{4.3}$$

where $\boldsymbol{\Sigma} = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix}$ is a matrix of EigenNails, and $\mathbf{w} = \begin{bmatrix}w_1 & w_2 & \cdots & w_k\end{bmatrix}^T$ is a vector of Nail Space coordinates. Multivariable linear least squares is applied to relate the force to the Nail Space coordinates using the model

$$\mathbf{F} = \mathbf{K}\mathbf{w} + \mathbf{F}_0 \tag{4.4}$$

where $\mathbf{K} = \begin{bmatrix}\mathbf{a} & \mathbf{b} & \mathbf{c}\end{bmatrix}^T$ is a matrix of coefficients, and $\mathbf{F}_0 = \begin{bmatrix}f_{x,0} & f_{y,0} & f_{z,0}\end{bmatrix}^T$ is the force offset vector. Once an image has been registered, it may be projected onto the EigenNails to obtain the Nail Space coordinates. Then, (4.4) may be applied to estimate the force.

4.4.3 AAM Parameters Models

Three potential models suggest themselves, based on the Shape, Texture, and Appearance parameters found during the formation of the Active Appearance Model (AAM) used to register the images. These parameters are similar in concept to the Nail Space coordinates found during formation of the EigenNail Magnitude model, in that each represents coordinates in a vector space spanned by the eigenvectors representing the variation of the data set. In the case of the Shape model, for example, the eigenvectors represent the modes of variation in the Shape of the finger contour and define a Shape Space, and so the Shape parameters correspond to the Shape Space coordinates.

Suppose that the Shape parameters ($\mathbf{b}_s$), Texture parameters ($\mathbf{b}_g$), and Appearance parameters ($\mathbf{c}$) for the calibration data set have been extracted. The Shape Parameters model is defined as

$$\mathbf{F} = \mathbf{S}\mathbf{b}_s + \mathbf{F}_0 \tag{4.5}$$

where the coefficient matrix $\mathbf{S} = \begin{bmatrix}\mathbf{a} & \mathbf{b} & \mathbf{c}\end{bmatrix}^T$ and the force offset vector $\mathbf{F}_0$ are identified using multivariable linear least squares. The Texture Parameters model ($\mathbf{F} = \mathbf{T}\mathbf{b}_g + \mathbf{F}_0$)

and the Appearance Parameters model ($\mathbf{F} = \mathbf{Ac} + \mathbf{F}_0$) function in exactly the same way, with a different set of parameters for each model. The coordinates for a given model are calculated by projecting a given image onto the corresponding eigenvectors, a process which takes place during image registration. With these coordinates, the force on any desired image may be estimated.

4.5 Results

The effects of the three modifications to the process are evaluated on multiple criteria as explained previously. An optimal combination of parameters is sought, which improves the performance of the force prediction. Unless otherwise specified, the force prediction model used is the EigenNail Magnitude Model. All other analyses use all data available (i.e., white and green LEDs, Cartesian and cylindrical grid data).

4.5.1 Lighting

Each subject experiences both green and white light on both fingers over the full range of forces. Each set of images from one combination of subject, finger, and LED color are registered using a different model.

4.5.1.1 Registration Error

The registration error, grouped by LED color and finger, is shown in Fig. 4.6. There is no significant difference in the registration accuracy between the two lighting conditions across the two fingers. This similarity indicates that the green LEDs may be used for collecting data without impacting the ability to register the images.

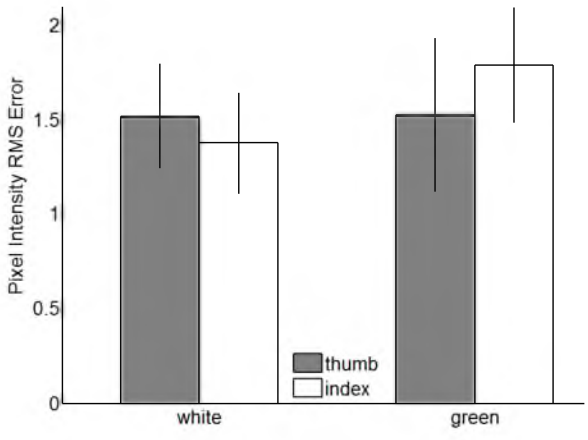

Fig. 4.6. Registration error grouped by LED color and finger. No significant difference is detected between the two lighting conditions across the two fingers.

4.5.1.2 Force Prediction Error

To estimate the force prediction error, a subset of the data is selected at random to form the model. For this work, 75 % of the data is selected to form the model. The remaining 25 % is presented to the resulting model for force prediction. This approach simulates an actual calibration phase followed by the acquisition of new image data. The validation error $\mathbf{e}_j$ for the jth image is calculated as the difference between the predicted force $\mathbf{F}_{p,j}$ and the measured force $\mathbf{F}_{m,j}$. The validation experiment is performed 100 times, randomly separating the data each time, and the mean validation error for each image is calculated to estimate the "true" validation error for each image. Finally, the overall RMS error for a data set is calculated.

The interaction effect between LED color and force direction on force prediction error is shown in Fig. 4.7. No significant difference is found between LED color across all force directions. Since [5] found that green light correlates better with force prediction accuracy, this result may require further investigation. The difference in results may be due to differences between the photodetectors used in previous studies and the camera CCD photodetectors.

4.5.2 Calibration Grids

The calibration grids will be evaluated on three criteria: (1) their effect on registration accuracy, (2) their effect on force prediction accuracy, and (3) the observability and identifiability of the corresponding calibration matrices. Registration error and force prediction accuracy were defined previously. Observability indicates which model parameters (or combinations of parameters) may not be able to be estimated given the current data

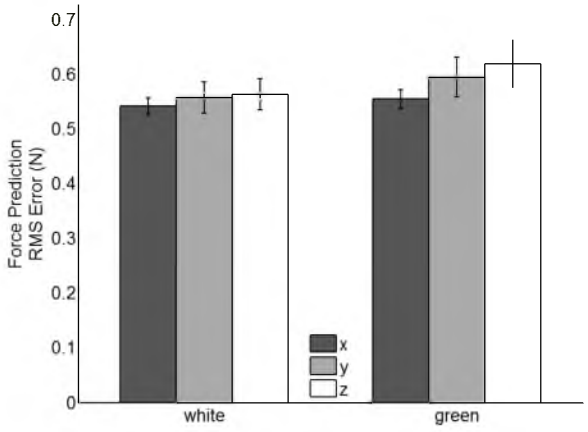

Fig. 4.7. Force prediction error grouped by LED color and force direction. No significant difference is detected between the LED colors among all three force directions.

set. Identifiability refers to the likelihood that any model parameters can be estimated successfully from the given calibration set.

4.5.2.1 Registration Error

The registration error is shown grouped by calibration grid and finger in Fig. 4.8. There is no significant difference between the two grids' effects on registration accuracy. Since the choice of force locations visited should have little or no effect on the registration process, this result is expected.

4.5.2.2 Force Prediction Error

ANOVA likewise demonstrates that there is no significant effect on force prediction error between the two calibration grids across all force directions (Fig. 4.9). While it has been assumed that such grids may have a task-specific effect, it appears that force prediction functions equally well between these two grid designs.

The number of images used to form a prediction model likewise has an effect on the force prediction error as well as the calibration time. If the same accuracy could be achieved using fewer images, less data would need to be collected, and the calibration time could be shortened. To determine the effect of the size of the data set on the force prediction error, the number of calibration images is reduced methodically to reduce the number of points by $\frac{1}{6}$, $\frac{1}{3}$, $\frac{1}{2}$, and $\frac{2}{3}$. Images are removed at random to achieve the desired number of points, which are 485, 388, 291, and 194, respectively. The validation experiments are then repeated, using only the EigenNail Magnitude Model. The results for all subjects experiencing the Cartesian grid, grouped by number of images and force direction, are

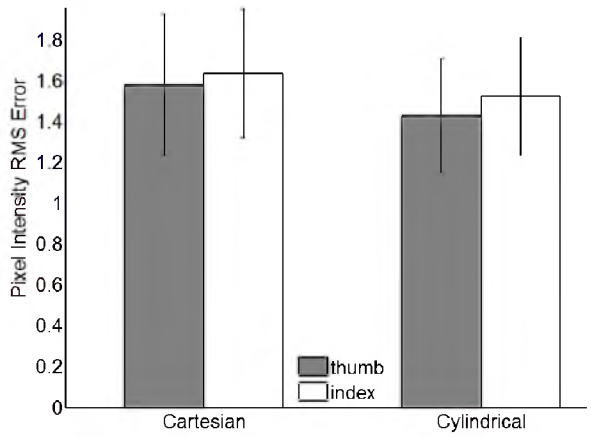

Fig. 4.8. Registration error grouped by calibration grid and finger. No significant difference is detected between the calibration grids across both fingers.

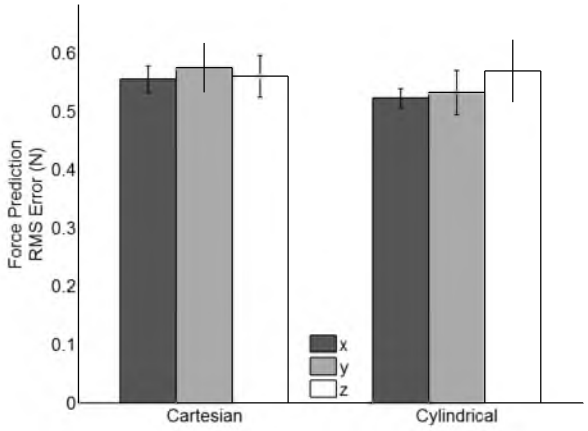

Fig. 4.9. Force prediction error grouped by calibration grid and force direction. No significant difference is detected between the two calibration grids across all force directions.

shown in Fig. 4.10. While the full grid, with 582 points, has an RMS error of 0.57 ± 0.01 N, each of the reduced-point grids has a significantly larger error, beginning with the 485-point grid (0.62 ± 0.02 N).

For the cylindrical grid, the standard data set contains 750 images. Thus, the reduced numbers of points are 625, 500, 375, and 250, respectively. The results for all subjects experiencing the cylindrical grid, grouped by number of images and force direction, are shown in Fig. 4.11. The full grid has an RMS error of 0.55 ± 0.02 N, and each of the reduced-point grids has a significantly larger error, beginning with the 625-point grid (0.60 ± 0.02 N).

Finally, the two grid types are shown together in Fig. 4.12. This illustrates that the force prediction accuracy is affected by the number of calibration images in the same pattern

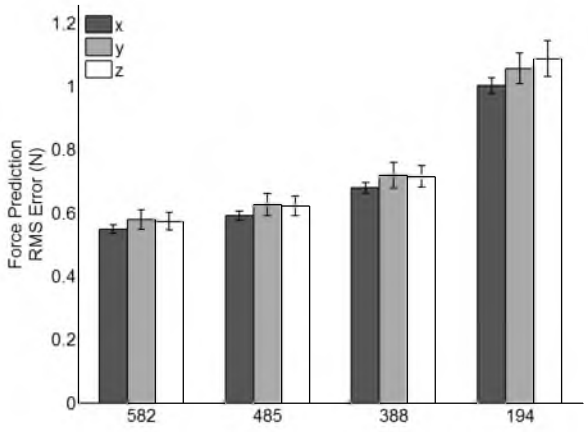

Fig. 4.10. Force prediction error grouped by Cartesian grid size and force direction. The prediction error is significantly different across all grid sizes, indicating that the standard number of images (582) achieves the most accurate results.

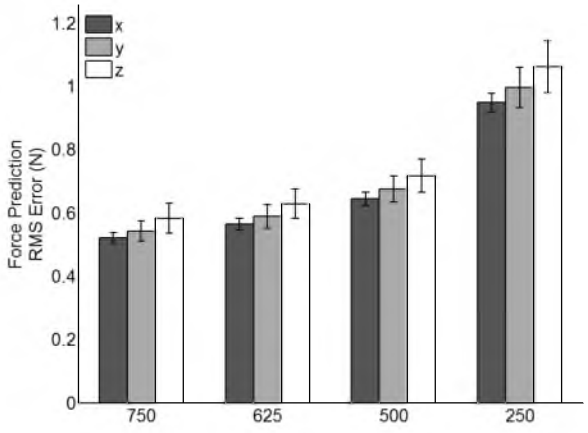

Fig. 4.11. Force prediction error grouped by cylindrical grid size and force direction. The prediction error is significantly different across all grid sizes, indicating that the standard number of images (750) achieves the most accurate results.

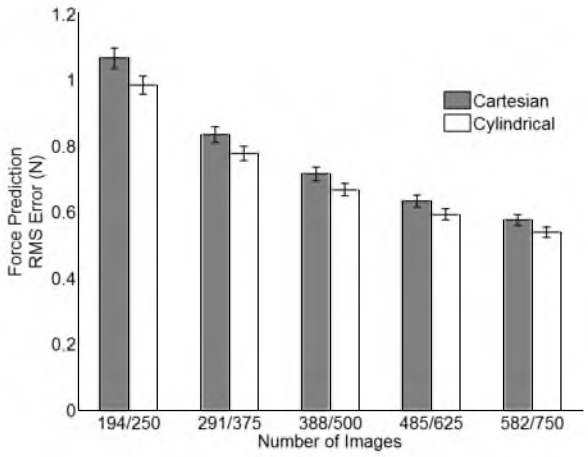

Fig. 4.12. Force prediction error grouped by grid size and grid type. While no difference is detected between Cartesian and cylindrical grids, the standard number of images results in the lowest force prediction error.

across both grids and shows that the increased number of images used in the cylindrical grid does not affect the general trend of the error.

4.5.2.3 Observability/Identifiability

The authors have chosen to use four observability criteria [80], defined as O_1 through O_4, to compare these two calibration grids. The formulas for the indices are

$$O_1 = \frac{(\mu_1 \mu_2 \cdots \mu_N)^{\frac{1}{N}}}{\sqrt{M \cdot P}} \tag{4.6}$$

$$O_2 = \frac{\mu_N}{\mu_1} \tag{4.7}$$

$$O_3 = \mu_N \tag{4.8}$$

$$O_4 = \frac{\mu_N^2}{\mu_1} \tag{4.9}$$

where μ_j is the jth of the N singular values, M is the number of equations per pose, and P is the number of poses recorded. Observability index O_1 is the root of the product of the N singular values. O_2 is the ratio of the smallest to the largest singular values, while O_3 is the minimum singular value. Observability index O_4 is the product of O_3 and O_2. Each of these indices indicates a better data set by a lower result.

The observability results are shown in Fig. 4.13. No significant difference is detected in observability between the two calibration trajectories using any of these four observability indices.

One identifiability criterion, the condition number $I = \frac{\mu_1}{\mu_N}$, is used. The range suggested by Schröer [70] indicates that the calibration model parameters are identifiable from a given calibration set if the condition number of the matrix is less than 100. The two grids have an average identifiability index (across all combinations of subject, finger, and LED color) of 74 ± 12 (white) and 70 ± 7 (green). No significant difference is detected between them. These values indicates that all model parameters are likely to be identifiable using both calibration trajectories.

4.5.3 Force Prediction Models

The five force prediction models are compared quantitatively on their ability to estimate forces. They are also evaluated qualitatively on the pixels used to form the model and those

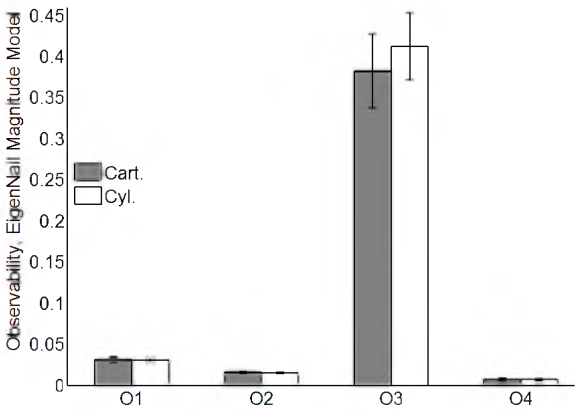

Fig. 4.13. Observability index grouped by calibration grid. Since each observability index is independent of the others, only the interaction between the two calibration grids for any given index is meaningful. For all indices, no statistically significant difference exists between the two grids.

pixels' influence on the final model.

4.5.3.1 Force Prediction Error

For the five prediction models, the force errors are aggregated by force direction as shown in Fig. 4.14. The EigenNail Magnitude Model performs significantly better than the other four models in all three directions of force, with an RMS error of approximately 0.55 ± 0.02 N in all three directions. The other four models exhibit errors varying from 0.65 ± 0.03 N to 2.8 ± 1.9 N. Somewhat surprisingly, the Texture Parameters Model, though seemingly equivalent to the EigenNail Magnitude Model in principle, has a much larger error. This may be due to the Texture Parameters Model being formed using only a few training images, or it may be due to the extensive scaling process used to condition the pixel intensities prior to formation of the final Texture model. The Linearized Sigmoid Model is significantly worse than all of the other models, a result that is unsurprising, considering that it generally rejects more than 80 % of the pixels for each finger. The Texture and Appearance Parameters models are statistically equivalent (which is intuitively appealing since the Texture parameters dominate in the formation of the Appearance model).

4.5.3.2 Qualitative Analysis

A secondary, qualitative check that may be performed is to create a set of synthetic images where each pixel's intensity indicates the contribution of that pixel in the prediction model to the corresponding direction of force. For the Linearized Sigmoid Model, the qualitative analysis images are straightforward. Each image is simply the corresponding

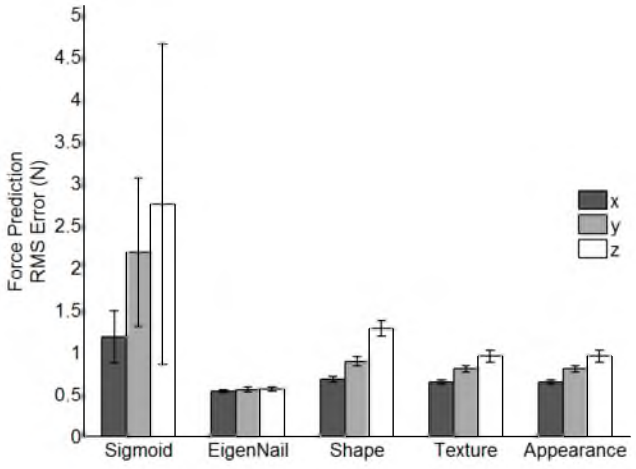

Fig. 4.14. Force prediction error grouped by prediction model and force direction. The EigenNail Magnitude Model has significantly lower force prediction error than all of the other models in all three directions.

column in the coefficient matrix ($\mathbf{p}_x = \mathbf{a}$, $\mathbf{p}_y = \mathbf{b}$, and $\mathbf{p}_z = \mathbf{c}$). The resulting images for one subject are shown in Fig. 4.15 and Fig. 4.16. Part (a) shows the pixel offset ($\mathbf{d}$), while parts (b)–(d) show the images corresponding to the forces in x, y, and z, respectively.

These images (and the following qualitative images for the other models) use yellow to represent positive correlation, or pixels that increase in intensity as the force increases, and green to represent negative correlation, pixels that decrease as the force decreases. Note that, due to the sign convention, decreasing normal force is equivalent to pressing more firmly with the finger. The patterns shown here correspond to those expected: the asymmetric pattern exhibited when the finger is under x-direction force, the opposing bands near the distal end and across the middle of the nail under y-direction force, and the band near the distal end of the finger paired with the discoloration in the finger along the sides of the nail under z-direction force all appear in the corresponding image maps. These same patterns are exhibited, even more clearly, in the following corresponding images.

It should be noted that these images show the major problem with the Linearized Sigmoid Model: the majority of the pixels cannot be used due to either insufficient linearity or variation, and so only a small fraction of the total area of the finger affects the results. In many cases, the pixels chosen by the algorithm are surprising and do not seem to be relevant to the blood flow within the finger. This deficiency is likely the reason for the much larger errors exhibited by this model.

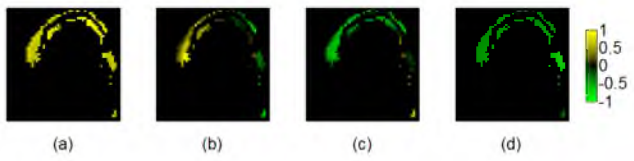

Fig. 4.15. Qualitative Linearized Sigmoid Model analysis (index). The intensity of each pixel in these images represents the contribution that pixel makes to (a) the pixel offset ($\mathbf{d}$), (b) the force prediction in each of the x-, (c) y-, and (d) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$.

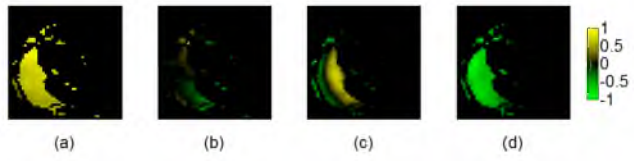

Fig. 4.16. Qualitative Linearized Sigmoid Model analysis (thumb). The intensity of each pixel in these images represents the contribution that pixel makes to (a) the pixel offset ($\mathbf{d}$), (b) the force prediction in each of the x-, (c) y-, and (d) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$.

For the EigenNail Magnitude Model, the formula used to generate these images is

$$\mathbf{p}_j = \sum_{i=1}^{k} a_{ji}\boldsymbol{\phi}_i\sqrt{\lambda_i} \tag{4.10}$$

where λ_i is the eigenvalue corresponding to the ith EigenNail, and the index j corresponds to the force direction (x, y, and z). The resulting index and thumb images for one subject are shown in Fig. 4.17 and Fig. 4.18, respectively. The expected patterns are exhibited in these images.

The qualitative images for the Texture Parameters Model are formed in a similar fashion. The formula is nearly identical, with the only differences being in the notation. The images are formed using the Texture eigenvectors $\boldsymbol{\phi}_g$, the Texture eigenvalues λ_g, and the Texture Space coordinate weights t_{ji}, combined in the same form as for the EigenNail Magnitude Model: $\mathbf{g}_j = \sum t_{ji}\boldsymbol{\phi}_{g,i}\sqrt{\lambda_{g,i}}$, where the index j again represents the x-, y-, or z- coordinate. The qualitative images for the same subject, for index finger and thumb, are shown in Fig. 4.19 and Fig. 4.20. In these images, the same patterns for F_x and F_y are shown.

The major apparent difference comes in the z-direction images, where the index fingernail is now mostly yellow with a green band near the distal end. While there are many potential causes of this phenomenon, it is important to recognize that the yellow is darker

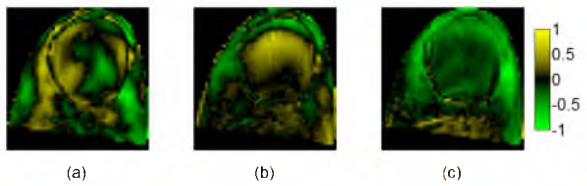

Fig. 4.17. Qualitative EigenNail Magnitude Model analysis (index). The intensity of each pixel in these images represents the contribution that pixel makes to the force prediction in each of the (a) x-, (b) y-, and (c) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$.

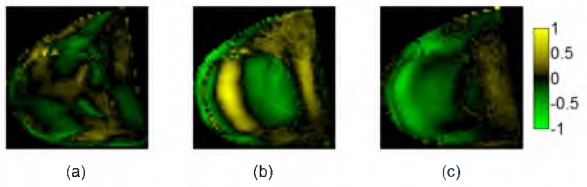

Fig. 4.18. Qualitative EigenNail Magnitude Model analysis (thumb). The intensity of each pixel in these images represents the contribution that pixel makes to the force prediction in each of the (a) x-, (b) y-, and (c) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$.

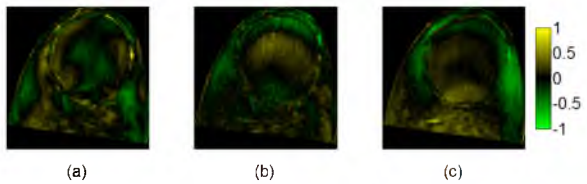

Fig. 4.19. Qualitative Texture Parameters Model analysis (index). The intensity of each pixel in these images represents the contribution that pixel makes to the force prediction in each of the (a) x-, (b) y-, and (c) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$.

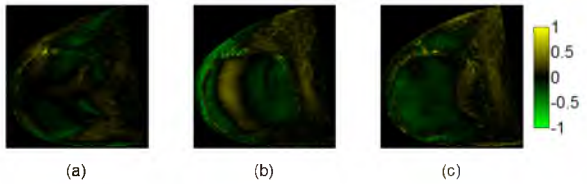

Fig. 4.20. Qualitative Texture Parameters Model analysis (thumb). The intensity of each pixel in these images represents the contribution that pixel makes to the force prediction in each of the (a) x-, (b) y-, and (c) z-directions. The brighter a pixel, the more it contributes. The images are each scaled to the range $(-1, 1)$.

(i.e, near zero), and the green in this area of the EigenNail Magnitude Model image is likewise near zero. Pixels in this region may be affected by the force in the z-direction either in the positive or negative direction, but the force does not have a large effect on the pixel intensity. Visual inspection of the finger confirms that this region of the finger does not change much under the influence of normal force. Thus, this result is to be expected.

The other result of note from these images is the few "bright spots" in the images and their locations. The locations of highest intensity in the images for the EigenNail Magnitude Model are in regions most closely associated with the color changes corresponding to that direction of force. In the images for the Texture Parameters Model, the most intense pixels are located in seemingly unimportant locations, such as the right side of the fingernail in the thumb or the skin just below the nail in the index finger. This result indicates that the Texture Parameters Model has trained the intensity of these pixels to be the most important when estimating force in these directions, which does not correspond to results of visual inspection of the fingers. Such a surprising effect may be due to the subset of Training Images being used to form the Texture model and thus not containing the entire range of Texture variation in the data set, or it may reflect the scaling of the intensity values that takes place in the formation of the Texture model. Either way, it is likely that this difference

is the cause of the poorer accuracy of the Texture Parameters Model.

4.6 Conclusion

This paper has investigated the optimization of the calibration and modeling of fingernail imaging for predicting fingernail force. A new force prediction model (the EigenNail Magnitude Model) was developed and shown to be capable of simultaneously estimating three-dimensional fingerpad force with an RMS error of 0.55 ± 0.02 N. In addition, two calibration grids were found to be equivalent in both their effect on registration accuracy and on force prediction error. Complementing these results is the determination that white LEDs and green LEDs have no significant effect on registration accuracy or force prediction error. Taken together, these results provide improvements to the automated calibration process.

The studies performed herein may be used to develop calibration and prediction techniques for use in measuring individual finger forces during human grasping experiments. Such measurements could be used to improve the ability of robots to analyze and duplicate human grasps. In medical studies, these techniques might be employed to characterize human development, or to track progression of cognitive conditions that affect motor skills.

CHAPTER 5

CONCLUSION

This dissertation has presented several improvements and expansions of fingernail imaging for force prediction. The major contributions are (1) an automated calibration system for collecting calibration images, (2) a new registration method that accounts for variation inherent in the finger and nail, and (3) a series of optimizations to identify the important variables within the calibration process. These contributions have been shown to reduce or maintain the force prediction RMS error of fingernail imaging across a variety of subjects.

With the automated calibration system in place, a test subject is no longer required to apply all of the desired forces manually. Instead, a force controller is applied to the the MLHD flotor so that it may sustain the desired forces with an RMS error of 0.05 N in all three directions. Given this controller, a test session lasting approximately 2 hours results in nearly 3000 images, where manual calibration limited data collection to around 300 images per subject. As discussed in Chapter 4, increasing the number of images reduces the force prediction error of the calibrated model. This method also allows the collection of force combinations that are difficult for an individuals to consciously control, such as those not on the edges of the friction cone. Although no quantitative work has been performed measuring the effect of using these forces in the model, this work prepares the way for such a study to be performed.

The new registration method, applying Active Appearance Models in an iterative manner, demonstrates several improvements over previous registration methods. It is faster and more accurate than other methods, without requiring parameter tuning, and eliminates the need to reject images due to registration problems. This method has been evaluated as the number of landmark points is reduced and found to preserve both registration and force prediction accuracy. In addition, the choice of color channel has been found to have no effect on registration accuracy, meaning the green channel could be used alone to register images. Both of these modifications will reduce registration time.

While prior force prediction models were limited to either predicting force on an image in a single direction or assigning an image to one of six direction groups using a classifier,

the new model developed here (the EigenNail Magnitude Model) has been shown to predict force with an RMS error of 0.55 ± 0.02 N in all three directions simultaneously. Four other models have been evaluated using similar methods and found to perform more poorly.

Two more parameters have been evaluated for their effects on registration and prediction. The choice of calibration grid type has no significant effect on either image registration or force prediction. However, the density of points in the grid does. No limit was detected in the data sets taken during this work, although it appeared that an asymptote was nearly reached. The study undertaken here could be used as a template if different calibration grids are designed in future work.

The color of lighting imposed on the finger during image collection also has no significant effect on image registration or force prediction. This surprising result needs to be investigated further as previous results indicate that green light should improve prediction performance. It is possible that the differences between this work and the conclusions found with fingernail sensors are due to variations in sensing equipment, differences in lighting, or it may be due to a fundamental difference in the way the two sensing methods transduce the force signal.

5.1 Future Work

Multiple avenues for further exploration exist. In the short term, the effect of variation between users in selecting the initial training image contours should be investigated. If the smoothing algorithm is not sufficient to remove the variation between these users, it will need to be improved.

It should be possible to make the registration process faster by automating the placement of the initial contours on training images, rather than asking users to select them manually. An initial procedure for this would involve a combination of edge-finding and joining of contours, including some resolution of problems along the same lines as those encountered when using the Canny Registration Method discussed in Chapter 1.

An alternative approach would be to build a much larger database of fingers (including hundreds of subjects) with only a few (approximately 10–15) training images from each subject. Then, an Active Appearance Model may be formed from all of these training images, and this model used to register any new image that arrives. Given enough training data, the model should be sophisticated enough to recognize most fingers presented to it.

The registration process may also be made faster by removing the `am_tools` programs (see Appendix B) from the loop. In the interests of time and efficiency, the author used existing applications to generate some of the intermediate results. However, this required

interrupting the workflow in the middle of the registration process, running a separate program, and then returning to the original workflow once the external program had completed. Integrating this step into the overall series of Matlab programs would improve the speed of the entire process.

These results depend on carefully controlled lighting with no specular reflection on the finger. While any such reflection on the finger would obscure the data beneath it, it may be possible to methodically detect and remove such imperfections in the data, using a method developed by Swaminathan et al. [81]. Such a result would be an important step to moving these experiments from the lab to real situations.

Finally, future work should apply all of these results to human grasping experiments. There are two major challenges yet to be overcome before such experiments may occur. First, during preliminary studies, it has been found that the fingers quickly move beyond the borders of the image or rotate so that the nail ceases to be nearly perpendicular to the stationary camera. It has been proposed to mount the cameras on 5-DOF robots and use the registration techniques developed here to track the finger and adjust the camera pose to maintain perpendicularity. Second, during these same preliminary studies, it was found that stationary lighting was insufficient to maintain consistent light on the finger. Two solutions suggest themselves: either mount the lights on separate robots following the same type of tracking algorithm, or use a series of spaced light sources, where different sources are assigned to shine at different intensities, depending on the finger pose. Once these two challenges are solved, a grasping experiment using fingernail imaging to measure the individual finger forces should be possible.

APPENDIX A

ADDITIONAL CONTROLLER DETAILS

This appendix contains material relating to the force controller that did not fit within the scope of the paper in Chapter 2.

A.1 General Equipment Notes

A.1.1 MLHD

The MLHD system consists of the stator (a stationary spherical base containing six permanent magnets and 3 two-dimensional optical sensors) and the flotor (an end effector containing six electromagnetic Lorentz coils and three LEDs). While the flotor is not mechanically connected to the stator, actuation is achieved by aligning the coils with the permanent magnets and generating current through the coils. By varying the currents, the position and orientation of the flotor is modified. Position and orientation sensing is performed by aligning the LEDs with the optical sensors.

As stated in Section 2.3.1, the flotor has a translational workspace of radius 12 mm and a rotational workspace of $\pm 8°$ in all three directions. It should be noted, however, that, just as the workspace is spherical rather than cubic, full simultaneous rotations in x, y, and z of $8°$ are not possible. In addition, the full range of translation is not available when extreme rotations are in effect. The entire range has not been explored, but it is also asymmetric. For example, simultaneous rotations in $+y$ and translations in $+x$ have a different limit than simultaneous rotations in $-y$ and translations in $-x$. It may be beneficial for future work to fully characterize these limits.

The system is also nonlinear in several different ways. The flotor is essentially an inverted pendulum in the rotational x and y directions. When the position controller in either of these directions is turned off, the flotor must be firmly grasped to maintain stability. In addition, it is known that a force command sent at one position does not produce the same result as a force command given at a different position. As with the workspace limits, these nonlinearities have not been fully characterized since the controller has been sufficient to compensate for them. However, it may be possible to design a more efficient controller in

the future, if the nonlinearities are identified.

Given the small workspace of the MLHD, sudden, relatively large changes in position and/or force often result in unstable operation and the flotor leaving the workspace. When the flotor leaves the workspace, at least one of the optical position sensors is generally no longer aligned. Consequently, the system must be manually realigned before calibration can resume. While realignment is not a difficult process, it requires training and familiarity with the system, and the process of "lift off" for the flotor freezes the client computer for approximately 5 seconds. Overall, it generally takes anywhere from 30 seconds to 2 minutes to recover from the flotor leaving the workspace. It would be beneficial to find a way to limit the number of times the flotor leaves the workspace.

It has been proposed that the open-loop force control available through the `setForce()` function in the MLHD API could be used to control the forces. However, this open-loop control assumes a perfectly accurate model of the electrical coil dynamics, the magnetic dynamics between permanent magnet and coil, and the mechanical translational/rotational dynamics of the flotor. Since the built-in compensation model can never be completely accurate and the flotor has been modified from its original dynamics, a closed-loop feedback controller is required to limit steady-state error and to provide desired system performance. The `setForce()` command is used to provide the wrench commands that control the flotor, but the values sent to the flotor are determined by the closed-loop controller.

A.1.2 Force Sensor

The Nano17 force sensor is used in concert with the ATI signal conditioning hardware as well as a custom-built low-pass filter designed to remove signals above 40 Hz. The voltages from the sensor are read using the analog inputs on a Sensoray S626 data acquisition card, using the 10 V transducing range. From there, the voltages (s_1 through s_6) are converted to forces and torques using the calibration equation supplied by the manufacturer:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} -0.0001 & 0.0638 & -0.1052 & 1.7541 & 0.0284 & -1.6721 \\ -0.0476 & -1.9468 & -0.0302 & 1.0996 & -0.0567 & 0.8878 \\ 1.8590 & 0.0072 & 1.9666 & -0.0315 & 1.7720 & -0.0150 \\ 0.3781 & 0.2180 & 10.6226 & -0.2446 & -10.3128 & -0.0379 \\ -11.8349 & -0.0240 & 6.8510 & 0.0358 & 5.4326 & -0.1363 \\ -0.0965 & -7.0361 & 0.4954 & -7.1050 & -0.0145 & -6.5864 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \end{bmatrix} \tag{A.1}$$

A.2 System Identification

The mass m_f and center of gravity $\mathbf{r}_f$ of the flotor are as defined in Chapter 2. The inertia matrix is defined in the frame of the flotor and with respect to the flotor axes, as

$$\mathbf{I}_f = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \tag{A.2}$$

The model of the flotor used here assumes a single, rigid body floating freely in space (three degrees of translational freedom and three degrees of rotational freedom) with a single wrench applied at some unknown point of action (O_f). The Newton–Euler equations for the flotor are

$$\begin{bmatrix} \mathbf{F}_f \\ \boldsymbol{\tau}_f \end{bmatrix} = \mathbf{H}(\boldsymbol{\Theta})\ddot{\boldsymbol{\Theta}} + \mathbf{V}\left(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}\right) + \mathbf{G}(\boldsymbol{\Theta}) \tag{A.3}$$

where the joint "angle" vector $\boldsymbol{\Theta} = \begin{bmatrix} \mathbf{p}^T & \boldsymbol{\theta}^T \end{bmatrix}^T$ includes both the position $\mathbf{p}$ and orientation $\boldsymbol{\theta}$ of the flotor. Using this notation, the joint angular velocity vector may be written as $\dot{\boldsymbol{\Theta}} = \begin{bmatrix} \mathbf{v}^T & \boldsymbol{\omega}^T \end{bmatrix}^T$ and the joint angular acceleration vector may be written as $\ddot{\boldsymbol{\Theta}} = \begin{bmatrix} \mathbf{a}^T & \boldsymbol{\alpha}^T \end{bmatrix}^T$. All positions and orientations are defined in the world frame. Each of the other terms is defined as

$$\mathbf{F}_f = \begin{bmatrix} F_{xm} \\ F_{ym} \\ F_{zm} \end{bmatrix} + {}^0\mathbf{R}_f \begin{bmatrix} F_{xs} \\ F_{ys} \\ F_{zs} \end{bmatrix} \tag{A.4}$$

$$\boldsymbol{\tau}_f = \begin{bmatrix} \tau_{xm} \\ \tau_{ym} \\ \tau_{zm} \end{bmatrix} + {}^0\mathbf{R}_f \begin{bmatrix} 0 \\ 0 \\ z_s \end{bmatrix} \times {}^0\mathbf{R}_f \begin{bmatrix} F_{xs} \\ F_{ys} \\ F_{zs} \end{bmatrix} \tag{A.5}$$

$$\mathbf{H}(\boldsymbol{\Theta}) = \begin{bmatrix} m_f I_3 & -m_f \mathbf{S}\left({}^0\mathbf{R}_f \mathbf{r}_f\right) \\ m_f \mathbf{S}\left({}^0\mathbf{R}_f \mathbf{r}_f\right) & {}^0\mathbf{R}_f \mathbf{I}_f {}^f\mathbf{R}_0 \end{bmatrix} \tag{A.6}$$

$$\mathbf{V}\left(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}\right) = \begin{bmatrix} m_f \mathbf{S}(\boldsymbol{\omega})\,\mathbf{S}(\boldsymbol{\omega})\,{}^0\mathbf{R}_f \mathbf{r}_f \\ \mathbf{S}(\boldsymbol{\omega})\,\mathbf{I}_f {}^0\mathbf{R}_f \boldsymbol{\omega} \end{bmatrix} \tag{A.7}$$

$$\mathbf{G}(\boldsymbol{\Theta}) = \begin{bmatrix} -m_f \mathbf{g} \\ -m_f \mathbf{S}\left({}^0\mathbf{R}_f \mathbf{r}_f\right) \mathbf{g} \end{bmatrix} \tag{A.8}$$

where F_{im} is the force exerted by the MLHD on the flotor in the i direction, F_{is} is the force exerted by the finger on the flotor in the i direction, τ_{im} is the torque exerted by the MLHD on the flotor in the i direction, I_3 is the 3×3 identity matrix, ${}^0\mathbf{R}_f$ is the rotation matrix between the flotor frame and the world frame, $\mathbf{g}$ is the gravity vector defined in the world frame, and $\mathbf{S}(\mathbf{x})$ is the skew-symmetric matrix "operator" on the vector $\mathbf{x}$. It is assumed that the finger does not exert pure torques on the flotor since the contact is approximately a point.

Figure A.1 repeats and enlarges Figure 2.5 for clarity. The derivation of the calibration equations are given in the following section, with the observability and identifiability of the parameters addressed in the following section.

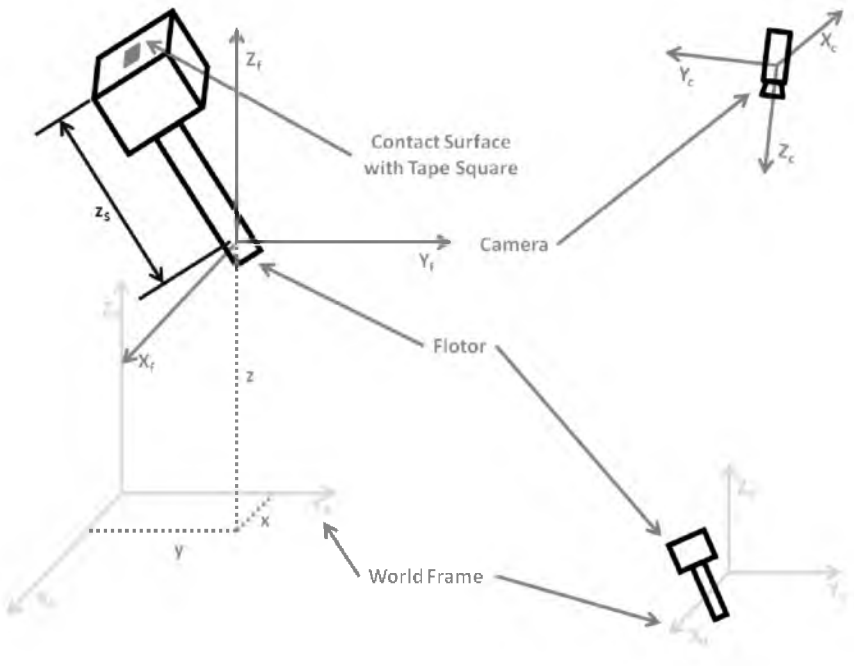

Fig. A.1. Enlarged Schematic of Kinematic Calibration. On the left is a close-up view of the flotor, showing the World Frame (X_0, Y_0, Z_0) in light gray, the flotor coordinates (x, y, z) and unrotated Flotor Frame (X_f, Y_f, Z_f) in dark gray, and the rotated flotor in black. The distance from the Flotor Frame origin to the contact surface z_s is indicated. On the right is shown the flotor and camera (both in black) relative to the World Frame (again in light gray), and the Camera Frame (X_c, Y_c, Z_c) in dark gray. During kinematic calibration, a small piece of tape provides fiducial marks on the contact surface, while the flotor moves through the workspace and the camera records images.

A.2.1 Static Parameter Identification

The flotor's axes are assumed to be parallel to the force sensor's axes. The contact surface, where the tape is attached, is assumed to be parallel to the flotor's xy-plane at a distance of z_s along the flotor's z-axis. The tape square's four corners have coordinates ${}^f\mathbf{p}_1 = \begin{bmatrix} x_1 & y_1 & z_s \end{bmatrix}^T$ through ${}^f\mathbf{p}_4 = \begin{bmatrix} x_4 & y_4 & z_s \end{bmatrix}^T$ in the frame of the flotor. These coordinates are transformed into the world frame using the formula

$$ {}^0\mathbf{p}_i = {}^0\mathbf{R}_f {}^f\mathbf{p}_i + \mathbf{d}_{0f} \tag{A.9} $$

where $\mathbf{d}_{0f} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ contains the translational coordinates of the flotor (defined in the world frame) and ${}^0\mathbf{R}_f = \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)$ is the rotation matrix relating the flotor coordinates to the world coordinates. In this work, $\mathbf{R}_k(\theta)$ indicates rotation about the k-axis by the angle θ.

The camera frame is translated by a vector $\mathbf{d}_{0c} = \begin{bmatrix} c_x & c_y & c_z \end{bmatrix}^T$ and rotated by three angles, θ_{cx}, θ_{cy}, and θ_{cz} relative to the world frame. For the purposes of this calibration,

the rotation matrix relating the world frame to the camera frame is defined as

$$ {}^{0}\mathbf{R}_{c} = \mathbf{R}_{z}\left(\theta_{cz}\right)\mathbf{R}_{y}\left(\theta_{cy}\right)\mathbf{R}_{x}\left(\theta_{cx}\right) \tag{A.10} $$

A point's world-frame coordinates may thus be transformed into the camera frame coordinates using the formula

$$ {}^{c}\mathbf{p}_{i} = {}^{0}\mathbf{R}_{c}^{T}{}^{0}\mathbf{p}_{i} + \mathbf{d}_{0c} \tag{A.11} $$

where ${}^{c}\mathbf{p}_{i} = \begin{bmatrix} x_{c,i} & y_{c,i} & z_{c,i} \end{bmatrix}^{T}$ contains the coordinates of the point in the camera frame. Given these camera-frame coordinates, the coordinates of the point in the image $\mathbf{u}_{i} = \begin{bmatrix} u_{i} & v_{i} \end{bmatrix}^{T}$ are found using the formulas

$$ u_{i} = \frac{f x_{c,i}}{z_{c,i}} - u_{0} \tag{A.12} $$

$$ v_{i} = \frac{f y_{c,i}}{z_{c,i}} - v_{0} \tag{A.13} $$

where $u_{0} = -511$, $v_{0} = -384$, and $f = -3600$ have been established using prior identification on the camera.

Given these equations, the partial derivatives required for nonlinear least squares (i.e., the Gauss–Newton method) may be calculated. The parameters to be identified are $\boldsymbol{\phi} = \begin{bmatrix} c_{x} & c_{y} & c_{z} & \theta_{cx} & \theta_{cy} & \theta_{cz} & z_{s} & x_{1} & y_{1} & x_{2} & y_{2} & x_{3} & y_{3} & x_{4} & y_{4} \end{bmatrix}^{T}$. The Jacobian $\mathbf{J}_{p,i} = \frac{\partial \mathbf{u}_{i}}{\partial \boldsymbol{\phi}}$ of the image coordinates with respect to the parameters is

$$ \mathbf{J}_{p,i} = \begin{bmatrix} \frac{\partial u_{i}}{\partial \boldsymbol{\phi}} \\ \frac{\partial v_{i}}{\partial \boldsymbol{\phi}} \end{bmatrix} \tag{A.14} $$

Substituting from (A.12) and (A.13) and rearranging

$$ \mathbf{J}_{p,i} = \begin{bmatrix} \frac{f}{z_{c,i}^{2}}\left(z_{c,i}\frac{\partial x_{c,i}}{\partial \boldsymbol{\phi}} - x_{c,i}\frac{\partial z_{c,i}}{\partial \boldsymbol{\phi}}\right) \\ \frac{f}{z_{c,i}^{2}}\left(z_{c,i}\frac{\partial y_{c,i}}{\partial \boldsymbol{\phi}} - y_{c,i}\frac{\partial z_{c,i}}{\partial \boldsymbol{\phi}}\right) \end{bmatrix} $$

$$ \mathbf{J}_{p,i} = \frac{f}{z_{c,i}^{2}} \begin{bmatrix} z_{c,i} & 0 & -x_{c,i} \\ 0 & z_{c,i} & -y_{c,i} \end{bmatrix} \begin{bmatrix} \frac{\partial x_{c,i}}{\partial \boldsymbol{\phi}} \\ \frac{\partial y_{c,i}}{\partial \boldsymbol{\phi}} \\ \frac{\partial z_{c,i}}{\partial \boldsymbol{\phi}} \end{bmatrix} $$

$$ \mathbf{J}_{p,i} = \mathbf{F}_{i}\mathbf{J}_{cp,i} \tag{A.15} $$

where $\mathbf{J}_{cp,i} = \frac{\partial {}^{c}\mathbf{p}_{i}}{\partial \boldsymbol{\phi}}$ is the Jacobian of the point coordinates in the camera frame with respect to the parameters. It may be partitioned for computational ease into three submatrices $\mathbf{J}_{cp,i} = \begin{bmatrix} \mathbf{J}_{cc,i} & \mathbf{J}_{cr,i} & \mathbf{J}_{tc,i} \end{bmatrix}$ since the parameters related to each submatrix may be considered as a group. $\mathbf{J}_{cc,i}$ relates the point coordinates in the camera frame to the coordinates of the

camera relative to world frame. $\mathbf{J}_{cr,i}$ relates the point coordinates in the camera frame to the rotational angles of the camera relative to world frame. $\mathbf{J}_{tc,i}$ relates the point coordinates in the camera frame to the tape corner coordinates in the flotor frame. Each of these Jacobians is considered in one of the following sections, followed by a discussion of the formation of the Gauss–Newton iteration problem.

A.2.1.1 Camera Position

The three camera position coordinates have similar effects on the Jacobian. Thus, the Jacobian $\mathbf{J}_{cc,i}$ of the camera coordinates (${}^c\mathbf{p}_i$) with respect to the coordinate vector of the camera ($\mathbf{d}_{0c}$) is

$$\mathbf{J}_{cc,i} = \frac{\partial {}^c\mathbf{p}_i}{\partial \mathbf{d}_{0c}} \tag{A.16}$$

$$\mathbf{J}_{cc,i} = \frac{\partial}{\partial \mathbf{d}_{0c}} \left({}^0\mathbf{R}_c^{T\,0}\mathbf{p}_i + \mathbf{d}_{0c} \right) \tag{A.17}$$

$$\mathbf{J}_{cc,i} = \frac{\partial}{\partial \mathbf{d}_{0c}} \left({}^0\mathbf{R}_c^{T\,0}\mathbf{p}_i \right) + \frac{\partial \mathbf{d}_{0c}}{\partial \mathbf{d}_{0c}} \tag{A.18}$$

Since ${}^0\mathbf{R}_c^T$ and ${}^0\mathbf{p}_i$ are independent of the camera position $\mathbf{d}_{0c}$, the derivative of the first term is identically zero. The second term is the identity matrix, and so the Jacobian is simply

$$\mathbf{J}_{cc,i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.19}$$

Since this Jacobian is constant in the tape corner coordinates, the combined Jacobian for all points is simply the individual Jacobian, stacked

$$\mathbf{J}_{cc} = \begin{bmatrix} \mathbf{I}_3 \\ \mathbf{I}_3 \\ \mathbf{I}_3 \\ \mathbf{I}_3 \end{bmatrix} \tag{A.20}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix.

A.2.1.2 Camera Rotational Parameters

The camera rotational parameters $\left(\boldsymbol{\theta}_c = \begin{bmatrix} \theta_{cx} & \theta_{cy} & \theta_{cz} \end{bmatrix}^T \right)$ are grouped in a similar fashion. The Jacobian $\mathbf{J}_{cr,i}$ of the point coordinates in the camera frame with respect to the three angles is

$$\mathbf{J}_{cr,i} = \frac{\partial {}^c\mathbf{p}_i}{\partial \boldsymbol{\theta}_c}$$

$$\mathbf{J}_{cr,i} = \begin{bmatrix} \frac{\partial {}^c\mathbf{p}_i}{\partial \theta_{cx}} & \frac{\partial {}^c\mathbf{p}_i}{\partial \theta_{cy}} & \frac{\partial {}^c\mathbf{p}_i}{\partial \theta_{cz}} \end{bmatrix} \tag{A.21}$$

If only the derivative with respect to the angle in the k direction is considered, the notation becomes slightly simpler. Substituting from (A.11)

$$\frac{\partial^c \mathbf{p}_i}{\partial \theta_{ck}} = \frac{\partial}{\partial \theta_{ck}} \left({}^0\mathbf{R}_c^T {}^0\mathbf{p}_i + \mathbf{d}_{0c} \right) \tag{A.22}$$

$$\frac{\partial^c \mathbf{p}_i}{\partial \theta_{ck}} = {}^0\mathbf{R}_c^T \frac{\partial^0 \mathbf{p}_i}{\partial \theta_{ck}} + \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{ck}} {}^0\mathbf{p}_i + \frac{\partial \mathbf{d}_{0c}}{\partial \theta_{ck}} \tag{A.23}$$

Since ${}^0\mathbf{p}_i$ and $\mathbf{d}_{0c}$ do not depend on the camera angles, the partial derivatives of these terms are always zero. Thus, the derivative may be written $\frac{\partial^c \mathbf{p}_i}{\partial \theta_{ck}} = \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{ck}} {}^0\mathbf{p}_i$. This may be substituted into the formula for the Jacobian to find

$$\mathbf{J}_{cr,i} = \begin{bmatrix} \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cx}} {}^0\mathbf{p}_i & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cy}} {}^0\mathbf{p}_i & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cz}} {}^0\mathbf{p}_i \end{bmatrix} \tag{A.24}$$

Each of the derivatives may be expanded further, by substituting from (A.10):

$$\frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{ck}} = \frac{\partial}{\partial \theta_{ck}} \left[\mathbf{R}_z(\theta_{cz}) \mathbf{R}_y(\theta_{cy}) \mathbf{R}_x(\theta_{cx}) \right] \tag{A.25}$$

$$\frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{ck}} = \frac{\partial \mathbf{R}_z(\theta_{cz})}{\partial \theta_{ck}} \mathbf{R}_y(\theta_{cy}) \mathbf{R}_x(\theta_{cx}) \tag{A.26}$$

$$+ \mathbf{R}_z(\theta_{cz}) \frac{\partial \mathbf{R}_y(\theta_{cy})}{\partial \theta_{ck}} \mathbf{R}_x(\theta_{cx}) \tag{A.27}$$

$$+ \mathbf{R}_z(\theta_{cz}) \mathbf{R}_y(\theta_{cy}) \frac{\partial \mathbf{R}_x(\theta_{cx})}{\partial \theta_{ck}} \tag{A.28}$$

Two of the three derivative terms in this final equation will be zero for any direction k. Eliminating the appropriate values, the partial derivatives with respect to each of the angles are

$$\frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cx}} = \mathbf{R}_z(\theta_{cz}) \mathbf{R}_y(\theta_{cy}) \frac{\partial \mathbf{R}_x(\theta_{cx})}{\partial \theta_{cx}} \tag{A.29}$$

$$\frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cy}} = \mathbf{R}_z(\theta_{cz}) \frac{\partial \mathbf{R}_y(\theta_{cy})}{\partial \theta_{cy}} \mathbf{R}_x(\theta_{cx}) \tag{A.30}$$

$$\frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cz}} = \frac{\partial \mathbf{R}_z(\theta_{cz})}{\partial \theta_{cz}} \mathbf{R}_y(\theta_{cy}) \mathbf{R}_x(\theta_{cx}) \tag{A.31}$$

These formulas may be used to find the columns of (A.24). The equation changes only slightly for each corner of the tape, and so the final Jacobian may be written as

$$\mathbf{J}_{cr} = \begin{bmatrix} \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cx}} {}^0\mathbf{p}_1 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cy}} {}^0\mathbf{p}_1 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cz}} {}^0\mathbf{p}_1 \\ \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cx}} {}^0\mathbf{p}_2 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cy}} {}^0\mathbf{p}_2 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cz}} {}^0\mathbf{p}_2 \\ \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cx}} {}^0\mathbf{p}_3 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cy}} {}^0\mathbf{p}_3 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cz}} {}^0\mathbf{p}_3 \\ \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cx}} {}^0\mathbf{p}_4 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cy}} {}^0\mathbf{p}_4 & \frac{\partial^0 \mathbf{R}_c^T}{\partial \theta_{cz}} {}^0\mathbf{p}_4 \end{bmatrix} \tag{A.32}$$

A.2.1.3 Tape Coordinates

The flotor frame coordinates of the first corner of the tape square (x_1, y_1, z_s) are likewise grouped. The Jacobian $\mathbf{J}_{tc,1}$ relating the coordinates of the first corner in the camera frame to the first point's coordinates in the flotor frame is found using the following procedure:

$$\mathbf{J}_{tc,1} = \frac{\partial^c \mathbf{p}_1}{\partial^f \mathbf{p}_1} \tag{A.33}$$

$$\mathbf{J}_{tc,1} = \frac{\partial}{\partial^f \mathbf{p}_1}\left({}^0\mathbf{R}_c^{T\,0}\mathbf{p}_1 + \mathbf{d}_{0c}\right) \tag{A.34}$$

$$\mathbf{J}_{tc,1} = \frac{\partial {}^0\mathbf{R}_c^T}{\partial^f \mathbf{p}_1}{}^0\mathbf{p}_1 + {}^0\mathbf{R}_c^T\frac{\partial {}^0\mathbf{p}_1}{\partial^f \mathbf{p}_1} + \frac{\partial \mathbf{d}_{0c}}{\partial^f \mathbf{p}_1} \tag{A.35}$$

Since ${}^0\mathbf{R}_c$ and $\mathbf{d}_{0c}$ are independent of ${}^f\mathbf{p}_1$, the first and last terms are identically zero. Expanding the remaining term using (A.9)

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^T\frac{\partial}{\partial^f \mathbf{p}_1}\left({}^0\mathbf{R}_f{}^f\mathbf{p}_1 + \mathbf{d}_{0f}\right) \tag{A.36}$$

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^T\left(\frac{\partial {}^0\mathbf{R}_f}{\partial^f \mathbf{p}_1}{}^f\mathbf{p}_1 + {}^0\mathbf{R}_f\frac{\partial {}^f\mathbf{p}_1}{\partial^f \mathbf{p}_1} + \frac{\partial \mathbf{d}_{0f}}{\partial^f \mathbf{p}_1}\right) \tag{A.37}$$

As before, the first and last terms may be eliminated since ${}^0\mathbf{R}_f$ and $\mathbf{d}_{0f}$ are independent of ${}^f\mathbf{p}_1$. The remaining term is merely the identity matrix:

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\frac{\partial {}^f\mathbf{p}_1}{\partial^f \mathbf{p}_1}$$

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \tag{A.38}$$

Note that the order of this portion of the parameter vector is different from what is given here. In the parameter vector, z_s is given first, followed by x_1 and y_1. To correct this, the final matrix needs to be re-ordered:

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{bmatrix} \tag{A.39}$$

This would represent the Jacobian for this section, if those three values were the only parameters. However, this section involves the other xy-coordinates $(x_2, y_2, x_3, y_3, x_4, y_4)$, which are not adequately represented. Thus, we augment the final matrix with a column of zeros for each of the six remaining parameters:

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{A.40}$$

$$\mathbf{J}_{tc,1} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_1 \tag{A.41}$$

The Jacobian is slightly different for each of the other corners since z_s needs to be repeated for each one.

$$\mathbf{J}_{tc,2} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.42}$$

$$\mathbf{J}_{tc,2} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_2 \tag{A.43}$$

$$\mathbf{J}_{tc,3} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.44}$$

$$\mathbf{J}_{tc,3} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_3 \tag{A.45}$$

$$\mathbf{J}_{tc,4} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.46}$$

$$\mathbf{J}_{tc,4} = {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_4 \tag{A.47}$$

These equations may be combined to form the full Jacobian for this section:

$$\mathbf{J}_{tc} = \begin{bmatrix} {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_1 \\ {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_2 \\ {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_3 \\ {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_4 \end{bmatrix} \tag{A.48}$$

.

A.2.1.4 Calibration Matrix Formation

To form the calibration matrix $\mathbf{J}_{p,j}$ for the jth image, the three Jacobian submatrices (A.20), (A.32), and (A.48) are substituted into (A.15).

$$\mathbf{J}_{p,j} = \mathbf{F}_j \begin{bmatrix} \mathbf{J}_{cc} & \mathbf{J}_{cr} & \mathbf{J}_{tc} \end{bmatrix} \tag{A.49}$$

$$\mathbf{J}_{p,j} = \mathbf{F}_j \begin{bmatrix} \mathbf{I}_3 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cx}}{}^0\mathbf{p}_1 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cy}}{}^0\mathbf{p}_1 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cz}}{}^0\mathbf{p}_1 & {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_1 \\ \mathbf{I}_3 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cx}}{}^0\mathbf{p}_2 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cy}}{}^0\mathbf{p}_2 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cz}}{}^0\mathbf{p}_2 & {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_2 \\ \mathbf{I}_3 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cx}}{}^0\mathbf{p}_3 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cy}}{}^0\mathbf{p}_3 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cz}}{}^0\mathbf{p}_3 & {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_3 \\ \mathbf{I}_3 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cx}}{}^0\mathbf{p}_4 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cy}}{}^0\mathbf{p}_4 & \frac{\partial {}^0\mathbf{R}_c^T}{\partial \theta_{cz}}{}^0\mathbf{p}_4 & {}^0\mathbf{R}_c^{T\,0}\mathbf{R}_f\mathbf{A}_4 \end{bmatrix} \tag{A.50}$$

where $\mathbf{F}_j$ is a matrix containing the camera parameters:

$$\mathbf{F}_j = \begin{bmatrix} \mathbf{F}_1 & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} \\ \mathbf{0}_{2,3} & \mathbf{F}_2 & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} \\ \mathbf{0}_{2,3} & \mathbf{0}_{2,3} & \mathbf{F}_3 & \mathbf{0}_{2,3} \\ \mathbf{0}_{2,3} & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} & \mathbf{F}_4 \end{bmatrix} \tag{A.51}$$

where $\mathbf{0}_{2,3}$ is a 2×3 matrix of zeros. Assuming that the jth image displays all four corners, the eight Gauss–Newton equations describing the image may be written in the form $\Delta\mathbf{U}_j = \mathbf{J}_{p,j}\Delta\boldsymbol{\phi}$, where

$$\Delta\mathbf{U}_j = \begin{bmatrix} u_1 & v_1 & u_2 & v_2 & u_3 & v_3 & u_4 & v_4 \end{bmatrix}^T \tag{A.52}$$

These equations are stacked for all m images, adjusting as needed for missing points in the images. The stacked equations have the form $\Delta\mathbf{U} = \mathbf{C}\Delta\boldsymbol{\phi}$. The calibration proceeds using standard Gauss–Newton iteration. The full results are given in Table A.1.

A.2.2 Identifiability

Questions may arise as to the suitability of the sample data for observing and identifying all of the parameters determined in Section 2.4.3. It has been suggested [70] that the condition number of the matrix $\mathbf{A}$ should be less than 100 as a reliable heuristic to indicate whether the singular values are large enough to indicate that the parameters are all identifiable.

For the static parameter data (Section 2.4.3.1) in Table A.1, the matrix condition number is 54. Since this is below 100, it is reasonable to assume that all of the parameters are identifiable from the data set.

For the dynamic parameter data collected in Section 2.4.3.2 and used to calculate the parameter values in Table 2.1, the matrix condition number is 4.4, well within the acceptable range, indicating that the parameters are all identifiable from the data.

A.2.3 Observability

The author is familiar with five different observability indices from the literature [80]. The first four direct the investigator to the best data set by maximizing an index, while the fifth indicates the best data set by a minimum. The formulas for the indices are

$$O_1 = \frac{(\mu_1\mu_2\cdots\mu_N)^{\frac{1}{N}}}{\sqrt{M \cdot P}} \tag{A.53}$$

$$O_2 = \frac{\mu_N}{\mu_1} \tag{A.54}$$

$$O_3 = \mu_N \tag{A.55}$$

$$O_4 = \frac{\mu_N^2}{\mu_1} \tag{A.56}$$

Table A.1. Flotor/Camera Static Parameters

Quantity	Value	Quantity	Value	Tape Corner Coordinates	
c_x	-1.3 mm	θ_{cx}	186°	(x_1, y_1)	$(-13.9, 14.7)$mm
c_y	2.5 mm	θ_{cy}	$-0.9°$	(x_2, y_2)	$(13.1, 15.0)$mm
c_z	280 mm	θ_{cz}	180°	(x_3, y_3)	$(13.6, -8.98)$mm
z_s	48.1 mm			(x_4, y_4)	$(-13.1, -9.48)$mm

$$O_5 = \sum_{j=1}^{N} \frac{1}{\mu_j^2} \tag{A.57}$$

where μ_j is the jth of the N singular values, M is the number of equations per pose, and P is the number of poses recorded. Observability index O_1 is the root of the product of the N singular values. O_2 is the ratio of the smallest to the largest singular values, while O_3 is the minimum singular value. Observability index O_4 is the product of O_3 and O_2, while O_5 is the trace of $\left(\mathbf{C}^T\mathbf{C}\right)^{-1}$. The five observability indices are shown for the static calibration matrix and the two dynamic matrices (calibration and validation) in Table A.2.

A.3 Controller Implementation

To investigate the gain and phase margins of the calibration system, frequency response procedures were used. The calibration system was commanded to apply sinusoidal forces (or angles) in one direction at a time while the remaining directions were held constant. For all directions except F_z, the oscillations occurred around zero, and the constant value was zero. For F_z, the oscillations occurred around -3 N, and the constant value was -3 N. All oscillations had an amplitude of either 1 N or 1°. Frequencies varied from 0.1 Hz to 100 Hz.

Bode plots for all three directions of force are shown in Fig. A.2. The gain margins were 6.7 in F_x and 3.0 in F_y. From the plot, it is evident that the phase delay in F_z does not cross $-180°$ over this frequency range. Since hardware and software limitations prevent exploration of frequencies above 100 Hz, these cannot be explored. It is assumed that the gain margin is at least 10 since the magnitude plot rises no higher than this after the initial downward plunge. The phase margins for the force directions are 171.3°, 38.9°, and 76.8°, respectively.

Bode plots for the three rotational directions are shown in Fig. A.3. The gain margins were 2.3 in θ_x, 1.8 in θ_y, and 16.7 in θ_z. The phase margins in the three directions are 8.0°, 5.8°, and 89.4°, respectively. These margins indicate that there is a range of gains for which the controller will be stable. Although large variations in the finger dynamics could cause instability, the worst effect of an unstable system will be the need to realign the flotor and

Table A.2. Observability of Calibration and Validation Trajectories

Trajectory	O_1	O_2	O_3	O_4	O_5
Static Calibration	0.013	0.019	0.034	0.0006	2.12
Calibration	2.61	0.255	832	212	4.26×10^-6
Validation	2.56	0.261	835	218	4.41×10^-6

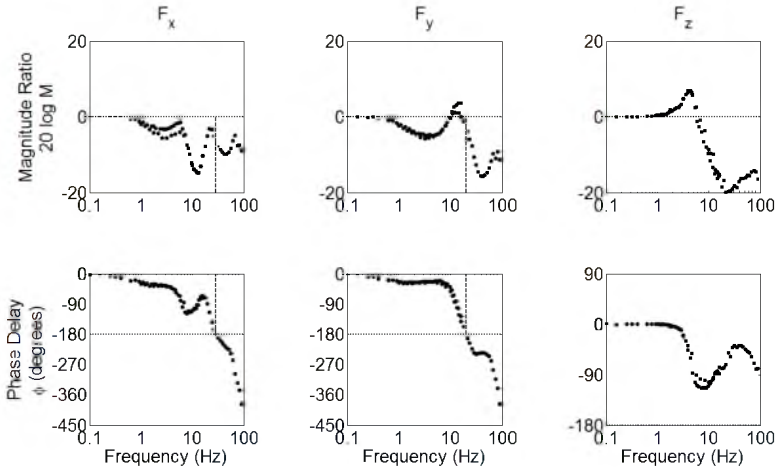

Fig. A.2. Bode Plots for F_x, F_y, and F_z. The magnitude plot is shown in the top row, with the phase plot shown in the bottom row. The x-axis in all plots contains (in log-scale) the frequency (in Hz) at which the sinusoid data were collected. In the magnitude plots, the y-axis gives $20 \log M$, or the gain of the response, with the horizontal dashed line indicating the zero-gain level. In the phase plot, the y-axis gives the phase delay (in degrees) of the response, with the horizontal dashed line indicating the -180° level. The vertical dashed line in the x- and y-direction plots represents the gain margin. No phase margin was found in the F_z plot.

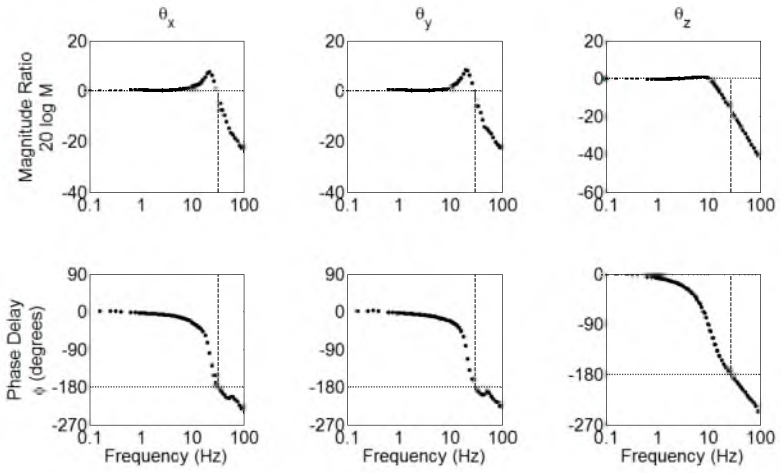

Fig. A.3. Bode Plots for θ_x, θ_y, and θ_z. The magnitude plot is shown in the top row, with the phase plot shown in the bottom row. The x-axis in all plots contains (in log-scale) the frequency (in Hz) at which the sinusoid data were collected. In the magnitude plots, the y-axis gives $20 \log M$, or the gain of the response, with the horizontal dashed line indicating the zero-gain level. In the phase plot, the y-axis gives the phase delay (in degrees) of the response, with the horizontal dashed line indicating the -180° level. The vertical dashed line in all plots indicates the gain margin in each direction.

restart the calibration (a procedure which requires less than one minute).

The closed-loop bandwidth of the force controller was also investigated. In the force directions, the bandwidth is 1.4 Hz (F_x), 1.4 Hz (F_y), and 7.0 Hz (F_z). For the angular directions, the bandwidth is 32.7 Hz (θ_x), 31.7 Hz (θ_y), and 12.8 Hz (θ_z).

APPENDIX B

ADDITIONAL REGISTRATION DETAILS

This appendix contains additional details pertaining to the registration method that did not fit within the scope of Chapter 3. Since that chapter is taken from a journal article that does not address differences in light color, only the 32 data sets (16 subjects, 2 fingers) collected under white LED light are used. However, this appendix pertains to the work as a whole and therefore contains all 64 data sets.

B.1 Alternative Registration Methods

An alternative method was proposed that involved a form of contour-matching, where the outside edge of the finger and the edge of the nail were fitted to ellipses, and these ellipses matched for registration. It was found that this method neglected many complexities of the finger, including the many nails that are not elliptical. It also had difficulty with noise in the image creating inappropriate edges, which caused incorrect identification of the contours. Finally, since this method used Canny edge finding to locate the edge points, it required the same parameter tuning as in the Canny method.

B.2 Training Images

The hand-selected contours for one training image from each combination of subject and LED color are shown in Fig. B.1 (index finger) and Fig. B.2 (thumb). Each finger is shown twice, once under white LED illumination (full RGB image) and once under green LED illumination (grayscale image). The finger contour is shown in red, while the nail contour is shown in yellow.

B.2.1 Smoothing the Contours

The first step of smoothing the contours, smoothing by linear interpolation, begins with calculating the Cartesian distance between each consecutive pair of points. A parameter, t, is designated as the cumulative distance from the beginning of the contour to the current point. Thus, the ith value of the parameter may be calculated as

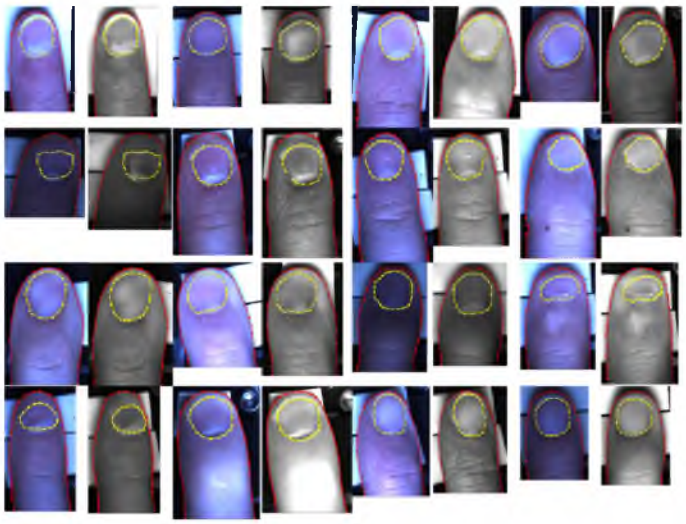

Fig. B.1. All subject/LED color combinations for the index finger, showing the user-selected contours. Each finger is shown twice, with the image under white LED illumination (full RGB image) on the left and the image green LED illumination (grayscale image) on the right. The finger contour is shown in red, while the nail contour is shown in yellow. These contours are later smoothed using the algorithm described in Chapter 3.

$$t_i = t_{i-1} + \Delta t_i \tag{B.1}$$

where $1 \leq i \leq m$, $t_1 = 0$ and

$$\Delta t_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{B.2}$$

For n desired points, the desired points are linearly interpolated across the interval $[t_1, t_m]$ using the formula

$$t_j = \frac{t_m}{m-1}(j-1) \tag{B.3}$$

for $1 \leq j \leq n$ and $t_m = t_n$. The coordinates (x_i, y_i) are then interpreted as functions of t and linearly interpolated using the standard formulas

$$x(t_j) = x_{i-1} + \frac{x_i - x_{i-1}}{t_i - t_{i-1}}(t_j - t_{i-1}) \tag{B.4}$$

$$y(t_j) = y_{i-1} + \frac{y_i - y_{i-1}}{t_i - t_{i-1}}(t_j - t_{i-1}) \tag{B.5}$$

for $t_{i-1} \leq t_j \leq t_i$.

When smoothing the outer finger edge contour, the first and last points selected terminate the contour and the desired number of points are used. When smoothing the nail edge contour, the first point is repeated at the end of the contour, so that smoothing may be performed over the entire closed loop. In addition, $n+1$ points are used, so the desired point count is still achieved, but the repeated endpoint is eliminated to avoid redundancy. An example of the process is shown in Fig. 3.8(a) for one finger contour and one nail contour.

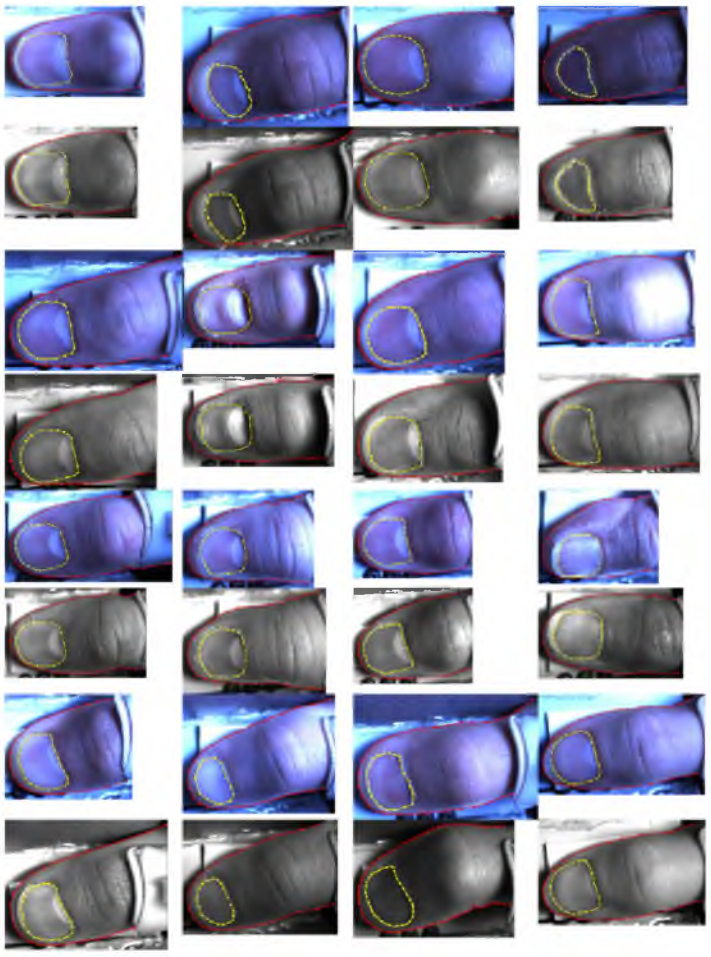

Fig. B.2. All subject/LED color combinations for the thumb, showing the user-selected contours. The thumbs are shown first under white LED illumination (full RGB image) and then once more under green LED illumination (grayscale image), below the first image. The finger contour is shown in red, while the nail contour is shown in yellow. These contours are later smoothed using the algorithm described in Chapter 3.

B.2.2 Adjusting the Nail Contour

The following formula is used to find the most distal nail point by interpolating the parallel coordinates:

$$x_{\parallel 0} = x_{\parallel 1} + \frac{x_{\parallel 2} - x_{\parallel 1}}{x_{\perp 2} - x_{\perp 1}} (-x_{\perp 1}) \tag{B.6}$$

B.2.3 Estimating the Angle of a Series of Points

Finding the angle a series of points makes with the positive x-axis is illustrated with an example data set, given in Fig. B.3. The actual difference between the maximum and minimum values for each direction is determined

$$\Delta x_a = \max \mathbf{x} - \min \mathbf{x} \tag{B.7}$$

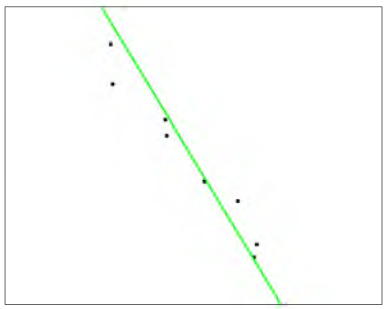

Fig. B.3. Estimating the Angle of a Series of Points. A total least squares fit line is found. The angle between this line and the positive x-axis is used as the angle of the series of points.

$$\Delta y_a = \max \mathbf{y} - \min \mathbf{y} \tag{B.8}$$

If $\Delta x_a > \Delta y_a$, then x is designated as the independent variable. Otherwise, y is used. Since the linear model used is the slope-intercept form, this step removes the possibility of an infinite (or near-infinite) slope.

Once the independent variable is designated, the total least squares procedure (see Section B.2.4) is used to fit the data to the linear model ($y_i = a_0 + a_1 x_i$ or $x_i = a_0 + a_1 y_i$). The difference between the maximum and minimum values of the fitted dependent variable is then found. Finally, the angle of the line is found using the two-argument arctangent function,

$$\theta = \operatorname{atan2}(\Delta y, \Delta x)$$

where the difference in the actual data is used for the independent variable, but the difference in the fitted data is used for the dependent variable.

B.2.4 Total Least Squares Fit of a Series of Points

Various methods of performing a total least squares fit have been explored in other sources [82, 83]. The procedure used in this work follows that given in [84]. The point coordinates (x_i, y_i) are first scaled by the standard deviation of the coordinates in each direction (σ_x, σ_y). This scaling reduces round-off error as well as bias towards the errors in one direction or the other:

$$x_{si} = \frac{x_i}{\sigma_x} \tag{B.9}$$

$$y_{si} = \frac{y_i}{\sigma_y} \tag{B.10}$$

This section assumes that x is the independent variable, although the two are interchangeable. The scaled data is assumed to fit a model with the equation $y_{si} = a_0 + a_1 x_{si}$.

The n data points may be recast as a matrix equation of the form $\mathbf{X}\boldsymbol{\phi} = \mathbf{Y}$, with $\boldsymbol{\phi} = \begin{bmatrix} a_1 & a_0 \end{bmatrix}^T$, $\mathbf{Y} = \begin{bmatrix} y_{s1} & y_{s2} & \cdots & y_{sn} \end{bmatrix}^T$ and

$$\mathbf{X} = \begin{bmatrix} x_{s1} & 1 \\ x_{s2} & 1 \\ \vdots & \vdots \\ x_{sn} & 1 \end{bmatrix} \tag{B.11}$$

The Singular Value Decomposition $\left(\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\right)$ of the augmented matrix $\mathbf{C} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} \end{bmatrix}$ is calculated. The $\mathbf{V}$ matrix is partitioned as follows:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & V_{22} \end{bmatrix} \tag{B.12}$$

where (in this case) $\mathbf{V}_{11}$ is 2×2, $\mathbf{V}_{12}$ is 2×1, $\mathbf{V}_{21}$ is 1×2, and V_{22} is a scalar. The total least squares fit for the parameters is

$$\boldsymbol{\phi}_{\text{TLS}} = -\frac{\mathbf{V}_{12}}{V_{22}} = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} \tag{B.13}$$

Once the line's parameters have been estimated, the parameters must be corrected to remove the scaling:

$$m = a_1 \frac{\sigma_y}{\sigma_x} \tag{B.14}$$

$$b = a_0 \sigma_y \tag{B.15}$$

The original data then fits the line $y = mx + b$ in a total least squares sense.

B.2.5 Final Smoothed Images

The initial (hand-selected) and final (smoothed) contours for one training image from each combination of subject and LED color are shown in Fig. B.4 (index finger) and Fig. B.5 (thumb). These correspond to the images shown in Fig. B.1 and Fig. B.2. As before, the intial finger contour is shown in red and the initial nail contour is shown in yellow. The smoothed contours are shown in black.

B.3 Shape Model

Further details and results of the Shape model formation are included here.

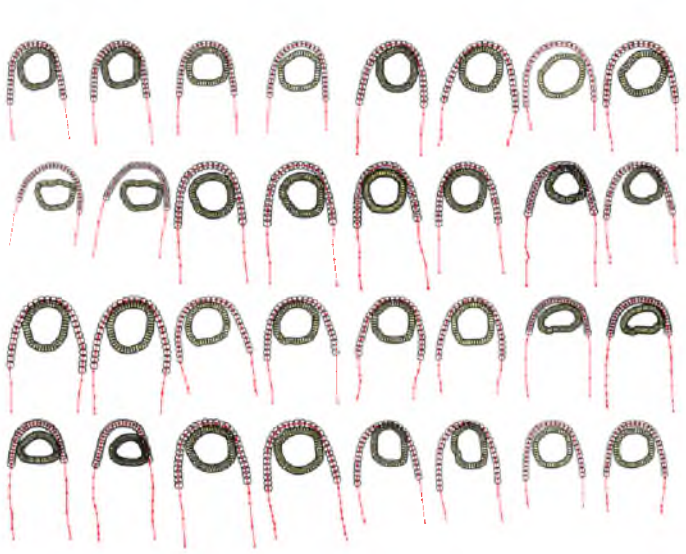

Fig. B.4. All subject/LED color combinations for the index finger, showing the user-selected contours and the final smoothed contours. Each contour corresponds to the image in the corresponding location from Fig. B.1. The finger contour is shown as red dots and lines, the nail contour is shown as yellow dots and lines, and the smoothed contour is shown as black circles.

B.3.1 Procrustes Alignment

First, for ease of notation, the Training Image shapes are extracted to column vectors. The ith shape vector has the form

$$\mathbf{x}_i = \begin{bmatrix} y_{1_i} \\ y_{2_i} \\ \vdots \\ y_{n_i} \\ x_{1_i} \\ x_{2_i} \\ \vdots \\ x_{n_i} \end{bmatrix} \tag{B.16}$$

(The y-coordinates are listed first, both for consistency with Cootes' work [39] and to have the row coordinates first, followed by the column coordinates.) The shape vectors are then aligned to the mean shape using Procrustes analysis [76, 77] within an iterative procedure as follows. The first shape is chosen as the initial estimate of the mean, and all other shapes are aligned to it. This involves finding the translation (t_x, t_y), scale $S(\mathbf{x})$, and rotation θ parameters that align two shapes, in the sense that the Procrustes distance between the first Training Image and each of the other images is minimized. The Procrustes distance P_d between two shapes $\mathbf{x}_1$ and $\mathbf{x}_2$ will be defined in (B.21). The translation parameters are simply the algebraic means of the coordinates:

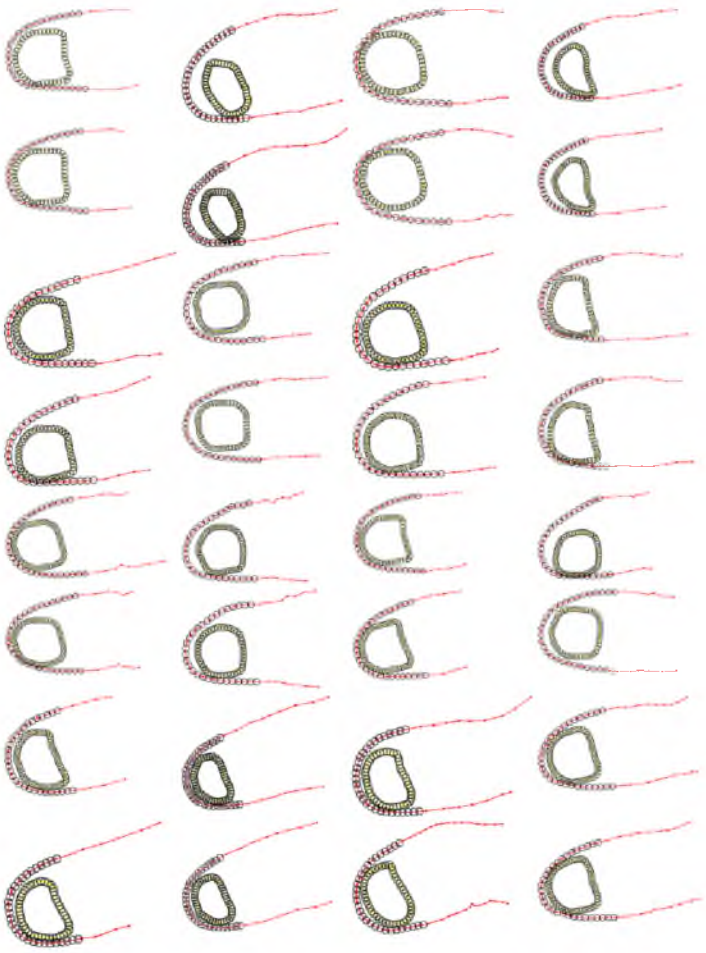

Fig. B.5. All subject/LED color combinations for the thumb, showing the user-selected contours and the final smoothed contours. Each contour corresponds to the image in the corresponding location from Fig. B.2. The finger contour is shown as red dots and lines, the nail contour is shown as yellow dots and lines, and the smoothed contour is shown as black circles.

$$t_x = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{B.17}$$

$$t_y = \frac{1}{n}\sum_{j=1}^{n} y_j \tag{B.18}$$

Scaling may be accomplished using any one of several scale metrics, although this work follows Stegmann's choice [75] to use the Frobenius norm,

$$S(\mathbf{x}) = \sqrt{\sum_{j=1}^{n}\left[(x_j - t_x)^2 + (y_j - t_y)^2\right]} \tag{B.19}$$

The rotation parameter is calculated in the following manner. First, the points are translated so that their centroids are at the origin. The coordinate values are divided by

the scale parameter and then rearranged into $n \times 2$ matrices. The matrix $\mathbf{B} = \mathbf{x}_1^T \mathbf{x}_2$ is found, and then the Singular Value Decomposition $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is calculated. Finally, the rotation matrix $\mathbf{R}$, which provides the optimal rotation from $\mathbf{x}_1$ to $\mathbf{x}_2$, is calculated as

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \mathbf{V}\mathbf{U}^T \tag{B.20}$$

For this work, the rotation parameter is calculated as $\theta = \text{atan2}\left(\mathbf{R}_{21}, \mathbf{R}_{11}\right)$.

The Procrustes distance between two shapes $\mathbf{x}_1$ and $\mathbf{x}_2$ is defined as

$$P_d^2 = \sum_{j=1}^{n} \left[(x_{j_1} - x_{j_2})^2 + (y_{j_1} - y_{j_2})^2 \right] \tag{B.21}$$

This value is calculated between each shape and the estimate of the mean. A new estimate of the mean shape, $\bar{\mathbf{x}}$, is calculated, where

$$\bar{\mathbf{x}} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{x}_i \tag{B.22}$$

where $\mathbf{x}_i$ is the shape vector from the ith of k Training Images. All of the shapes are then aligned to this new estimate of the mean, and the new Procrustes distance ($P_{d,\text{new}}$) is calculated. The relative percent error is calculated as

$$\%\text{Error} = \max \left\| \frac{P_{d,\text{new}} - P_d}{P_{d,\text{new}}} \right\| \times 100\% \tag{B.23}$$

The process is repeated until the percent error is less than some desired metric (in this case, 0.01 %).

Fig. B.6 illustrates, using the shapes from one subject's Training Images, how the entire process works. The initial set of shapes (a) are individually compared to the first shape (b). The alignment parameters are found and each shape is aligned, one by one, to the first shape (c). The mean shape is then calculated, and the process is repeated, until convergence (d).

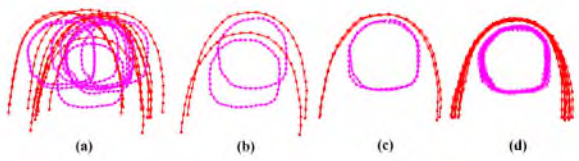

Fig. B.6. Shape alignment by Procrustes analysis. (a) Shapes from training images. (b) Two shapes to be aligned. (c) The same two shapes after alignment. (d) All training shapes after alignment.

B.3.2 Principal Component Analysis

The Shape Model covariance matrix is formed using the formula:

$$\mathbf{\Sigma}_s = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{B.24}$$

This matrix may be inspected to verify that the points are correlated, and Fig. B.7(a) shows a graphical representation of the covariance matrix for the training images of one subject's index finger. In this image, green pixels represent pairs of coordinates that are more negatively correlated, while yellow pixels represent pairs that are more positively correlated, and black pixels represent pairs of coordinates with low correlation. In a covariance matrix where the points are completely uncorrelated, the image would be entirely black except for a strongly yellow diagonal. The red lines are only for reference and divide the image into convenient regions representing the x- and y-coordinates of the finger and nail.

Further information may be gleaned from the correlation matrix, $\mathbf{\Gamma}$, which is formed by normalizing the covariance matrix by the variance

$$\mathbf{\Gamma} = \mathbf{V}\mathbf{\Sigma}_s\mathbf{V} \tag{B.25}$$

$$\mathbf{V} = \begin{bmatrix} \frac{1}{\sigma_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_n} \end{bmatrix} \tag{B.26}$$

The correlation matrix for the same subject/finger combination is shown in Fig. B.7(b). As with part (a), yellow indicates coordinate pairs that are positively correlated, green

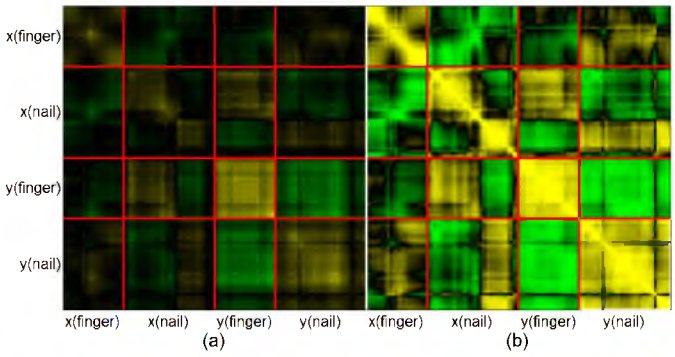

Fig. B.7. Shape covariance and correlation matrices. (a) Covariance and (b) Correlation matrices of Shape variation for one subject. In both figures, green pixels represent more negatively correlated pairs of coordinates, while yellow pixels represent more positively correlated pairs. Black pixels in both images represent pairs of coordinates with low correlation. The x- and y-coordinates of both the finger and the nail contours are correlated, indicating that they move in a synchronized manner.

indicates negatively correlated coordinate pairs, and black pixels have low correlation. This is essentially the same image with the contrast improved, which better illustrates the correlations and shows that the point coordinates are definitely not independent.

The number of eigenvalues and eigenvectors are limited using the following procedure: t modes are chosen to retain 99% of the variation in the training set, using the formula,

$$\sum_{i=1}^{t} \lambda_i \leq 0.99 \sum_{i=1}^{2n} \lambda_i \tag{B.27}$$

For most subjects, $t = 6$ or $t = 7$, although of the 64 data sets, one has $t = 5$ and three have $t = 8$. Fig. B.8 shows a plot of the eigenvalues (as a percentage of the total) for all subjects. This shows the typical behavior of the finger image sets, in that the first few eigenvalues contain most of the variation, followed by a few more to reach the 99 % level specified.

B.4 Texture Model

The ith texture vector, consisting of the m pixels raster-scanned from the aligned ith image, has the form

$$\mathbf{g}_i = \begin{bmatrix} p_{1i} \\ p_{2i} \\ \vdots \\ p_{mi} \end{bmatrix} \tag{B.28}$$

Normalizing the Textures is an iterative process, much the same as aligning the Shapes. To begin, the mean Texture $\bar{\mathbf{g}}$ is estimated as

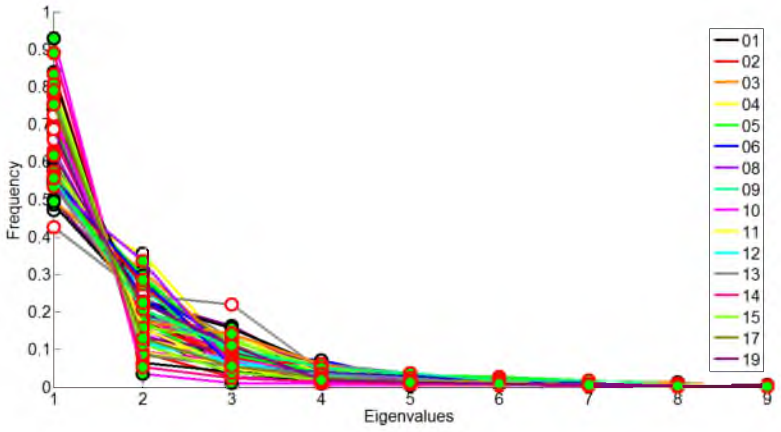

Fig. B.8. Shape eigenvalues for all subject/finger/LED color combinations. The first nine eigenvalues are represented along the x-axis, while the y-axis shows the relative weight of each eigenvalue for each combination. Line colors correspond to subjects, as shown in the legend. Circle edge colors (black vs. red) correspond to finger (thumb vs. finger). Circle face colors (white and green) correspond to LED color.

$$\hat{\bar{\mathbf{g}}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{g}_i \tag{B.29}$$

where each of the N Training Images is represented by its texture vector, $\mathbf{g}_i$. The mean vector estimate is then "standardized" using the formula

$$\bar{\mathbf{g}} = \frac{\hat{\bar{\mathbf{g}}} - \beta}{\alpha} \tag{B.30}$$

where α and β are the standard deviation and mean of $\hat{\bar{\mathbf{g}}}$. The Training Images are then "aligned" to $\bar{\mathbf{g}}$:

$$\mathbf{g}_{i,n} = \frac{\mathbf{g}_i - \beta_i}{\alpha_i} \tag{B.31}$$

$$\alpha_i = \mathbf{g}_i^T\bar{\mathbf{g}} \tag{B.32}$$

$$\beta_i = \frac{1}{m}\sum_{j=1}^{m} g_{ji} \tag{B.33}$$

This is an iterative process, in that the mean vector $\bar{\mathbf{g}}$ is estimated again after all of the Training Images have been aligned. The alignment and re-estimation of the mean are repeated until convergence is achieved.

The covariance and correlation matrices for the Texture model could be inspected in a fashion similar to that used for the Shape model. However, due to their size (tens of thousands of pixels) and less obvious structure (adjacent pixels in the image may be far from each other once the images have been raster scanned into pixel vectors), such an analysis would not be as useful. Thus, this qualitative exercise is not performed here.

As before, (B.27) is used to find the required number of eigenvectors to achieve 99 % of the variation. The same patterns of variation in the number of eigenvectors are present, with t being between 5 and 8. Fig. B.9 shows a plot of the eigenvalues as a percentage of the total, for all subject. The typical Texture eigenvalue plot shows a gradual reduction instead of the sharp dropoff seen in the Shape Model. This is likely due to the greater nuance present in the pixel intensity information.

B.5 Appearance Model

Since the covariance and correlation matrices are again of manageable size and have a recognizable structure, they may be inspected to determine the qualitative behavior of the data. Fig. B.10 shows both matrices. As before, green pixels show negative correlation, while yellow shows positive correlation, and black pixels are pairs with no correlation. From this figure, it may be seen that the Shape Parameters have no correlation among themselves,

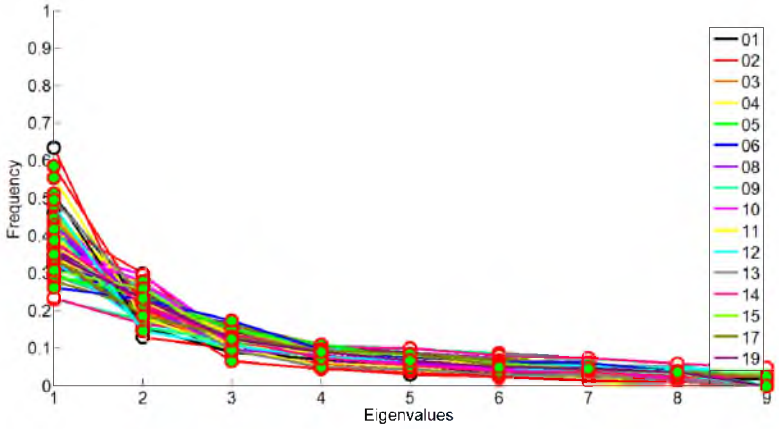

Fig. B.9. Texture eigenvalues for all subject/finger/LED color combinations. The first nine eigenvalues are represented along the x-axis, while the y-axis shows the relative weight of each eigenvalue for each combination. Line colors correspond to subjects, as shown in the legend. Circle edge colors (black vs. red) correspond to finger (thumb vs. finger). Circle face colors (white and green) correspond to LED color.

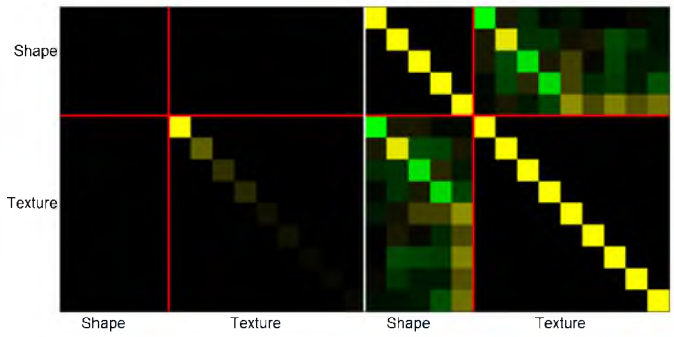

Fig. B.10. Appearance covariance and correlation matrices. (a) Covariance and (b) correlation matrices of Appearance variation for one subject. In both figures, green pixels represent more negatively correlated pairs of coordinates, while yellow pixels represent more positively correlated pairs. Black pixels in both images represent pairs of coordinates with low correlation.

and the Texture Parameters likewise have no intercorrelation. However, there is correlation between the Shape and Texture Parameters. This is to be expected since the color patterns are known to relate to force and the shape of the finger also relates to the force.

Again, (B.27) is used to find the required number of eigenvectors to achieve 99 % of the variation. As with the other models, between five and eight eigenvectors are identified for each subject. Fig. B.11 shows a plot of the eigenvalues as a percentage of the total, for all subjects. As with the Texture model, the dropoff is more gradual than with the Shape model, probably due to the Appearance model incorporating both types of information.

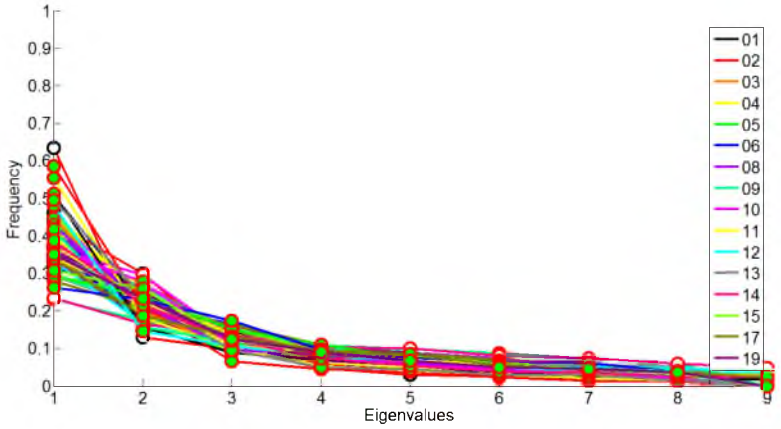

Fig. B.11. Appearance eigenvalues for all subject/finger/LED color combinations. The first nine eigenvalues are represented along the x-axis, while the y-axis shows the relative weight of the eigenvalue for each combination. Line colors correspond to subjects, as shown in the legend. Circle edge colors (black vs. red) correspond to finger (thumb vs. finger). Circle face colors (white and green) correspond to LED color.

B.6 Pixel Intensity Error

Fig. B.12 shows the mean RMS errors in pixel intensites by finger and LED color, across all subjects. Neither the thumb (1.5 ± 0.3) nor the index finger (1.4 ± 0.3) is found to have a statistically significant effect on the registration accuracy.

B.7 Method Modifications

B.7.1 Landmark Point Density

The method of Landmark Point reduction for a sample finger contour are illustrated in Fig. B.13. The endpoints of the finger contour are used as static points, and the contour is interpolated with the desired number of points. The initial nail contour point is retained, and the desired number of contour points is used to interpolate along the length. The original points are shown by dots, with "x" marks showing the new landmark points. An "o" marks the three points that remained in the same location.

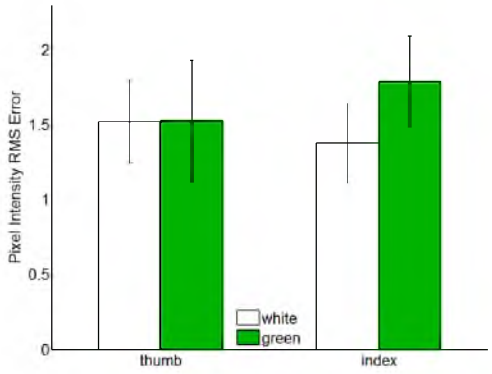

Fig. B.12. Errors in Pixel Intensity for each Finger/LED Color. The 95 % confidence intervals, calculated across all 16 subjects, are also shown.

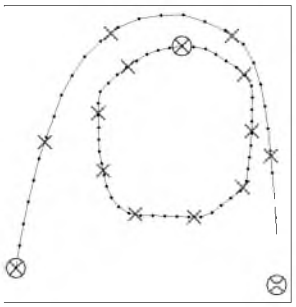

Fig. B.13. Landmark Point Reduction. The endpoints of the finger contour (marked with an "o") are retained, and the desired number of finger contour landmark points are interpolated along the length of the finger contour (marked with an "x"). Similarly, the most distal nail contour point (marked with an "o") is retained, and the desired number of nail contour landmark points are interpolated along the length of the nail contour (marked with an "x"). For reference, the original landmark points are marked with dots.

APPENDIX C

ADDITIONAL MODELING AND OPTIMIZATION DETAILS

This appendix contains additional details that did not fit within the scope of Chapter 4.

C.1 Calibration Grids

Between 582 and 750 images of each digit are recorded under each lighting condition. As explained in Chapter 4, these images are taken at specified locations distributed around the force space.

The space is divided into nine regions, numbered 1–9, as shown in Fig. C.1. Region 1 is the central core of shear force, with magnitude $F_s \leq 0.5\,\mathrm{N}$. Regions 2–9 are 45°-wide wedge-shaped regions containing nonzero shear forces. For example, region 2 spans the range $F_s > 0.5\,\mathrm{N}$ with angle $-157.5° \leq \theta < -112.5°$.

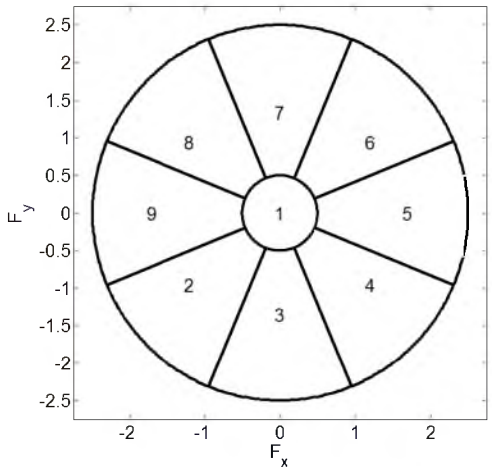

Fig. C.1. Regions of the calibration grid, viewed in the xy-plane. Region 1 contains forces near zero shear, while regions 2–9 are wedge-shaped nonzero shear forces arranged around the central core. During a typical calibration session, each region is visited 3 times, in a random order.

Each combination of subject/finger/light color was randomly assigned either a Cartesian or Cylindrical grid, as shown in Table C.1. This table also gives the order in which the nine trajectory regions were presented to each subject.

Table C.1. Grid Choice and Region Order for Each Data Set.

		LED Color		
Subj	Finger	White	Green	Region Order
	Thumb	Cart	Cyl	144982165528396269377437185
1	Index	Cyl	Cyl	82365254899497137524617 3186
	Thumb	Cyl	Cyl	18614225382678943314769 5597
2	Index	Cyl	Cart	21495568623947778549633 2118
	Thumb	Cart	Cyl	94174687841532538919223 6576
3	Index	Cart	Cyl	51869427123878157394369 4526
	Thumb	Cyl	Cyl	81125799384142633258469 6757
4	Index	Cart	Cart	23856426469239911838757 1547
	Thumb	Cart	Cart	69544928417116598675832 2337
5	Index	Cart	Cart	56847796839823671292514 4513
	Thumb	Cart	Cart	73965914851828365297346 2174
6	Index	Cyl	Cyl	79296114172366385538574 4289
	Thumb	Cart	Cart	39468625294786837115374 9521
8	Index	Cyl	Cyl	84518833647562412673291 9759
	Thumb	Cart	Cart	59919341627321884536456 7782
9	Index	Cart	Cyl	39484485732525362769187 1619
	Thumb	Cyl	Cart	32917626524459938718685 3471
10	Index	Cyl	Cyl	72764399539527328148615 6841
	Thumb	Cart	Cart	36695147564439128858271 2379
11	Index	Cyl	Cart	74656321989748851622315 9473
	Thumb	Cyl	Cart	81318259266753834475719 4962
12	Index	Cart	Cart	16159922864872955843437 7613
	Thumb	Cart	Cyl	96446257823187554263983 7911
13	Index	Cart	Cart	29268453477159486769311 3582
	Thumb	Cart	Cyl	51828971494126734393658 7652
14	Index	Cyl	Cart	73622546817115546233984 9978
	Thumb	Cyl	Cart	53975691781714246282649 3385
15	Index	Cyl	Cart	84281123557968469742961 3735
	Thumb	Cyl	Cart	68425884912533135176924 6797
17	Index	Cart	Cyl	57676843975811293594631 8224
	Thumb	Cyl	Cart	83986182579426135593421 6477
19	Index	Cyl	Cart	97812453353662196598477 4812

REFERENCES

[1] S. A. Mascaro and H. H. Asada, "Photoplethysmograph fingernail sensors for measuring finger forces without haptic obstruction," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 698–708, 2001.

[2] ——, "The common patterns of blood perfusion in the fingernail bed subject to fingertip touch force and finger posture," *Haptics-e: Electron. J. Haptics Res.*, vol. 4, no. 3, pp. 1–6, 2006.

[3] Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "Predicting fingertip forces by imaging coloration changes in the fingernail and surrounding skin," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 10, pp. 2363–2371, 2008.

[4] Y. Sun, "Measure fingertip force by imaging the fingernail and the surrounding skin," Ph.D. dissertation, University of Utah, 2007.

[5] J. M. Abu-Khalaf and S. A. Mascaro, "Effects of wavelength and optical path length in design of fingernail touch force sensing," *IEEE Sensors J.*, vol. 13, no. 2, pp. 807–815, 2013.

[6] J. M. Abu-Khalaf, J. W. Park, D. J. Mascaro, and S. A. Mascaro, "Stretchable fingernail sensors for measurement of fingertip force," in *3rd Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2009, pp. 625–626.

[7] M. C. F. Castro and A. Cliquet, Jr., "A low-cost instrumented glove for monitoring forces during object manipulation," *IEEE Trans. Rehabil. Eng.*, vol. 5, no. 2, pp. 140–147, 1997.

[8] Q. Fu, W. Zhang, and M. Santello, "Anticipatory planning and control of grasp positions and forces for dexterous two-digit manipulation," *J. Neurosci.*, vol. 30, no. 27, pp. 9117–9126, Jul. 2010.

[9] S. A. Winges, S. E. Eonta, J. F. Soechting, and M. Flanders, "Effects of object compliance on three-digit grasping," *J. Neurophysiol.*, vol. 101, p. 24472458, May 2009.

[10] K. N. Tarchanidis and J. N. Lygouras, "Data glove with a force sensor," *IEEE Trans. Instrum. Meas.*, vol. 52, pp. 984–989, Mar. 2003.

[11] B. B. Edin, G. Westling, and R. S. Johansson, "Independent control of human finger-tip forces at individual digits during precision lifting," *J. Physiol.*, vol. 450, pp. 547–564, 1992.

[12] J. F. Soechting and M. Flanders, "Sensorimotor control of contact force," *Curr. Opin. Neurobiol.*, vol. 18, pp. 565–572, 2008.

[13] S. E. Tomlinson, R. Lewis, and M. J. Carré, "Review of the frictional properties of finger-object contact when gripping," *Proc. Inst. Mech. Eng., Part J: J. Eng. Tribology*, vol. 221, no. 8, pp. 841–850, 2007.

[14] M. C. F. Castro and A. Cliquet, Jr., "Artificial grasping system for the paralyzed hand," *Artif. Organs*, vol. 24, no. 3, pp. 185–188, 2000.

[15] L. M. Muratori, T. L. McIsaac, A. M. Gordon, and M. Santello, "Impaired anticipatory control of force sharing patterns during whole-hand grasping in parkinson's disease," *Exp. Brain Res.*, vol. 185, no. 1, pp. 41–52, 2008.

[16] A. M. Gordon, "Task-dependent deficits during object release in parkinson's disease," *Exp. Neurol.*, vol. 153, no. 2, pp. 287–298, 1998.

[17] P. Raghavan, M. Santello, A. M. Gordon, and J. W. Krakauer, "Compensatory motor control after stroke: An alternative joint strategy for object-dependent shaping of hand posture," *J. Neurophysiol.*, vol. 103, no. 6, pp. 3034–3043, 2010.

[18] J. Monzée, Y. Lamarre, and A. M. Smith, "The effects of digital anesthesia on force control using a precision grip," *J. Neurophysiol.*, vol. 89, no. 2, pp. 672–683, 2003.

[19] A.-S. Augurelle, A. M. Smith, T. Lejeune, and J.-L. Thonnard, "Importance of cutaneous feedback in maintaining a secure grip during manipulation of hand-held objects," *J. Neurophysiol.*, vol. 89, no. 2, pp. 665–671, 2003.

[20] M. Santello and J. F. Soechting, "Force synergies for multifingered grasping," *Exp. Brain Res.*, vol. 133, no. 4, pp. 457–467, 2000.

[21] H. Han and S. Kawamura, "Analysis of stiffness of human fingertip and comparison with artificial fingers," in *Proc. IEEE Conf. Syst. Man Cybern.*, 1999, pp. 800–805.

[22] D. Pawluk and R. Howe, "Dynamic contact of the human fingerpad against a flat surface," *J. Biomech. Eng.*, vol. 121, no. 6, pp. 605–611, 1999.

[23] H. Han, A. Shimada, and S. Kawamura, "Analysis of friction on human fingers and design of artificial fingers," in *Proc. IEEE Intl. Conf. Robot. Autom.*, 1996, pp. 3061–3066.

[24] S. Marcheschi, F. Salsedo, M. Fontana, F. Tarri, O. Portillo-Rodriguez, and M. Bergamasco, "High performance explicit force control for finger interaction haptic interface," in *2nd Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2007, pp. 464–469.

[25] Y. Lin and Y. Sun, "5-D force control system for fingernail imaging calibration," in *Proc. IEEE Intl. Conf. Robot. Autom.*, 2011, pp. 1374–1379.

[26] T. Grieve, L. Lincoln, Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "3D force prediction using fingernail imaging with automated calibration," in *IEEE Symp. Haptic Interfaces Virtual Environment Teleoperator Syst.*, 2010, pp. 113–120.

[27] Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "Estimation of fingertip force direction with computer vision," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1356–1369, Dec. 2009.

[28] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147–151.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[30] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[31] Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "Eigennail for finger force direction recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3251–3256.

[32] N. Sugita, D. Iwai, and K. Sato, "Touch sensing by image analysis of fingernail," in *SICE Annu. Conf.*, 2008, pp. 1520–1525.

[33] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2001, pp. 511–518.

[34] S. Garg, A. Kumar, and M. Hanmandlu, "Biometric authentication using finger nail surface," in *12th Int. Conf. Intell. Syst. Des. Applicat.*, 2012, pp. 497–502.

[35] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[36] T. R. Grieve, J. M. Hollerbach, and S. A. Mascaro, "Fingernail image registration using active appearance models," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3011–3018.

[37] ——, "Force prediction by fingernail imaging using active appearance models," in *5th Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2013, pp. 181–186.

[38] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. 5th European Conf. Comput. Vision*, vol. 2, 1998, pp. 484–498.

[39] ——, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[40] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, "FAME - a flexible appearance modelling environment," *IEEE Trans. Med. Imag.*, vol. 22, no. 10, pp. 1319–1331, 2003.

[41] P. Tresadern, M. Ionita, and T. Cootes, "Real-time facial feature tracking on a mobile device," *Int. J. Comput. Vision*, vol. 96, no. 3, pp. 280–289, 2013.

[42] T. Neogi, M. Bowes, J. Niu, K. De Souza, G. Vincent, J. Goggins, Y. Zang, and D. T. Felson, "MRI-based three-dimensional bone shape of the knee predicts onset of knee osteoarthritis: Data from the Osteoarthritis Initiative," *Arthritis Rheum.*, vol. 65, no. 8, pp. 2048–2058, 2013.

[43] D. Kang, J. Woo, P. J. Slomka, D. Dey, G. Germano, and C.-C. J. Kuo, "Heart chambers and whole heart segmentation techniques: review," *J. Electron. Imaging*, vol. 21, no. 1, pp. 010 901–1–010 901–16, 2012.

[44] M. Roberts, T. F. Cootes, and J. E. Adams, "Vertebral morphometry: Semiautomatic determination of detailed shape from dual-energy x-ray absorptiometry images using Active Appearance Models," *Invest. Radiol.*, vol. 41, no. 12, pp. 849–859, 2006.

[45] M. G. Roberts, J. Graham, and H. Devlin, "Improving the detection of osteoporosis from dental radiographs using active appearance models," in *IEEE Int. Symp. Biomed. Imaging: From Nano to Macro*, 2010, pp. 440–443.

[46] ——, "Image texture in dental panoramic radiographs as a potential biomarker of osteoporosis," *IEEE Trans. Biomed. Eng.*, vol. PP, no. 99, pp. 1–1, 2013.

[47] M. G. Roberts, T. Oh, E. Pacheco, R. Mohankumar, T. F. Cootes, and J. E. Adams, "Semi-automatic determination of detailed vertebral shape from lumbar radiographs using Active Appearance Models," *Osteoporosis Int.*, vol. 23, no. 2, pp. 655–664, 2013.

[48] V. Kajić, M. Esmaeelpour, B. Považay, D. Marshall, P. L. Rosin, and W. Drexler, "Automated choroidal segmentation of 1060 nm OCT in healthy and pathologic eyes using a statistical model," *Biomed. Opt. Express*, vol. 3, no. 1, pp. 86–103, 2012.

[49] X. Chen, J. K. Udupa, U. Bagci, Y. Zhuge, and J. Yao, "Medical image segmentation by combining Graph Cuts and Oriented Active Appearance Models," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2035–2046, 2012.

[50] R. Cai, Q. Wu, R. Zhang, L. Fan, and C. Ruan, "Red blood cell segmentation using Active Appearance Model," in *11th IEEE Int. Conf. Signal Process.*, vol. 3, 2012, pp. 1641–1644.

[51] S.-W. Kim, J.-Y. Jung, S.-J. Lee, A. W. Morales, and S.-J. Ko, "Sensor fusion-based people counting system using the Active Appearance Models," in *IEEE Int. Conf. Consumer Electron.*, 2013, pp. 65–66.

[52] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski, "Affectaura: an intelligent system for emotional memory," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 849–858.

[53] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, "Painful monitoring: Automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database," *Image Vision Comput.*, vol. 30, no. 3, pp. 197–205, 2012.

[54] B.-J. Theobald and I. Matthews, "Relating objective and subjective performance measures for AAM-based visual speech synthesis," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 8, pp. 2378–2387, 2012.

[55] J. Suo, X. Chen, S. Shan, W. Gao, and Q. Dai, "A concatenational graph evolution aging model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2083–2096, 2012.

[56] A. W. Moore, J. Schneider, and K. Deng, "Efficient locally weighted polynomial regression predictions," in *Proc. Int. Mach. Learning Conf.*, 1997, pp. 236–244.

[57] Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "Measuring fingertip forces by imaging the fingernail," in *Proc. 14th Symp. Haptic Interfaces Virtual Environment Teleoperator Syst.*, 2006, pp. 125–131.

[58] "Diagonal 6mm (type 1/3) progressive scan CCD image sensor with square pixel for color cameras," ICX204AK, Sony Corporation.

[59] S. A. Mascaro and H. H. Asada, "Measurement of finger posture and three-axis fingertip touch force using fingernail sensors," *IEEE Trans. Robot. Autom.*, vol. 20, pp. 26–35, 2004.

[60] R. Gulati and M. Srinivasan, "Human fingerpad under indentation I: Static and dynamic force response," in *ASME Bioeng. Conf.*, vol. 29, 1995, pp. 261–262.

[61] P. Berkelman and R. Hollis, "Lorentz magnetic levitation for haptic interaction: device design, function and integration with simulated environments," *Int. J. Robot. Res.*, vol. 9, pp. 644–667, 2000.

[62] *Magnetic Levitation Haptic Interface User Manual*, CMU, Pittsburgh, PA, 2008.

[63] B. J. Unger, R. L. Klatzky, and R. L. Hollis, "Teleoperation mediated through magnetic levitation: Recent results," in *IEEE Conf. Mechatronics Robot.*, 2004, pp. 1453–1457.

[64] J. M. Abu-Khalaf and S. A. Mascaro, "Optimization of fingernail sensing technique based on optical experimentation and modeling," in *IEEE Sensors Applicat. Symp.*, Feb. 2011, pp. 283–288.

[65] P. W. Brand and M. H. Anne, *Clinical Mechanics of the Hand.* St. Louis: Mosby-Year Book, Inc., 1993.

[66] Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "Dynamic features and prediction model for imaging the fingernail to measure fingertip forces," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2813–2818.

[67] T. Grieve, Y. Sun, J. M. Hollerbach, and S. A. Mascaro, "3-D force control on the human fingerpad using a magnetic levitation device for fingernail imaging calibration," in *3rd Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2009, pp. 411–416.

[68] M. G. John Hollerbach, Wisama Khalil, *Model Identification.* Berlin; Heidelberg: Springer, 2008, pp. 321–344.

[69] C. Lawson and R. Hanson, *Solving Least Squares Problems.* Englewood Cliffs: Prentice Hall, 1974.

[70] K. Shroer, "Theory of kinematic modelling and numerical procedures for robot calibration," in *Robot Calibration*, R. Bernhardt and S. Albright, Eds. San Francisco: Morgan Kaufmann, 1990.

[71] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators.* London; New York: Springer, 2000.

[72] J. Liao and M. A. Srinivasan, "Experimental investigation of frictional properties of the human fingerpad," MIT Touch Lab, MA, Tech. Rep., 1999.

[73] C. Pylatiuk, A. Kargov, S. Schulz, and L. Doderlein, "Distribution of grip force in three different functional prehension patterns," *J. Med. Eng. Technol.*, vol. 30, no. 3, p. 176182, 2006.

[74] M. P. Rearick, A. Casares, and M. Santello, "Task-dependent modulation of multi-digit force coordination patterns," *J. Neurophysiol.*, vol. 89, pp. 1317–1326, Mar. 2003.

[75] M. B. Stegmann, "Active appearance models: Theory, extensions & cases," Master's thesis, Technical University of Denmark, 2000.

[76] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. Roy. Stat. Soc.*, vol. 53, no. 2, pp. 285–339, 1991.

[77] F. L. Bookstein, "Landmark methods for forms without landmarks: localizing group differences in outline shape," in *Proc. Workshop Math. Methods Biomed. Image Anal.*, 1996, pp. 279–289.

[78] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[79] W. F. Massy, "Principal components regression in exploratory statistical research," *J. American Stat. Assoc.*, vol. 60, no. 309, pp. 234–256, 1965.

[80] Y. Sun and J. M. Hollerbach, "Observability index selection for robot calibration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 831–836.

[81] R. Swaminathan, S. B. Kang, R. Szeliski, A. Criminisi, and S. K. Nayar, "On the motion and appearance of specularities in image sequences," in *Proc. 7th European Conf. Comput. Vision*, vol. 1, 2002, pp. 508–523.

[82] P. P. N. de Groen, "An introduction to total least squares," *Nieuw Archief voor Wiskunde, 4th Series*, vol. 14, pp. 237–253, 1996.

[83] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing.* Cambridge, UK: Cambridge University Press, 2007, pp. 785–788.

[84] I. Markovsky and S. V. Huffel, "Overview of total least-squares methods," *Signal Process.*, vol. 87, no. 10, pp. 2283–2302, Oct. 2007.