NAIST-IS-DD0661027

Doctoral Dissertation

Model Calibration and Unscented Kalman Filter for Hand Pose Estimation

Albert Causo

September 3, 2010

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Albert Causo

Thesis Committee:

Professor Tsukasa Ogasawara	(Supervisor)
Professor Naokazu Yokoya	(Co-supervisor)
Associate Professor Jun Takamatsu	(Co-supervisor)
Professor Etsuko Ueda	(Nara National College of Technology)

Model Calibration and Unscented Kalman Filter for Hand Pose Estimation[*]

Albert Causo

Abstract

Advances in vision-based hand pose estimation and gesture recognition improves human-robot interaction and human-computer interaction by allowing users to move more naturally, since they are unencumbered by wires and gadgets. Vision-based model-based approaches use cameras and hand models to estimate the hand pose. However, a survey of state-of-the-art in this field highlights three areas for improvement: model calibration, quantitative evaluation, and flexibility. The goal of this thesis is to address these three problems by calibrating the hand model specific to a user and applying predictive filtering.

Improving the quality of the hand model vis-a-vis the individual users and the need of the system translates to improvement in estimation accuracy. However, most works in this field either use data from expensive machines like MRI or adjust the model parameters along with pose estimation. To enable any user to use the system regardless of gender or physical differences such as hand size, hand model individualization using multiple cameras is proposed. The first part of the thesis presents a technique in calibrating the hand model prior to pose estimation. From the calibration motion, the method estimates the finger link lengths as well as the hand shape by minimizing the gap between the hand model and the observation. The feasibility of the proposal is confirmed by comparing actual and estimated link lengths. The performance of using calibrated model in the hand pose estimation method is compared against using uncalibrated hand model and against dataglove measurements; this allowed a quantitative evaluation

[*] Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0661027, September 3, 2010.

of the method. Results showed that pose estimation improves when the calibrated model is used. The motion profile of the calibrated model is much nearer to the actual hand's. Additionally, the resulting hand shape after pose estimation using the calibrated model is more similar to the actual hand shape than when using the uncalibrated model.

Majority of research find it necessary to fix either the global pose parameter or the local pose parameters with respect to the camera in order to accomplish pose estimation. This limits the flexibility of systems to accommodate natural hand motion. To address this issue Unscented Kalman Filter (UKF), a Bayesian-based filter, is applied in the hand pose estimation. UKF, a filter for non-linear systems, is chosen because the hand motion can be expressed as a non-linear problem. The filter attempts to minimize the gap between the hand model and the observation data. Initially, the UKF was tested using the uncalibrated model, a skeletal model covered with quadric surface skin; the observation data used was the voxel of the hand. Estimation results of different hand motions of up to 15DOF (global and local parameters) confirm the feasibility of the system. The use of calibrated model with UKF was also feasible but issues such as filter stability and computation speed have to be addressed to realize the full potential of the filter.

This thesis aims to improve hand pose estimation by addressing three major shortcomings of most pose estimation systems through calibration of the hand model and application of predictive filtering. Results show that pose estimation improvements are achieved and quantitative comparison with other pose acquisition system are possible by calibrating the hand model for each user. A more natural hand motion input is also made possible by simultaneously estimating global and local pose parameters with the use of UKF.

Keywords:

vision-based hand pose estimation, hand model calibration, Unscented Kalman Filter, multiple viewpoint cameras

モデルキャリブレーションとアンセンテッド・カルマンフィルタによる手姿勢推定[*]

アルベルト　カウソ

内容梗概

ビジョンベースの手姿勢推定やジェスチャ認識の進歩は，非接触な入力を可能とし，より自然なヒューマンロボットインタラクションやヒューマンコンピュータインタラクションを実現する．手姿勢を推定するためにカメラと手モデルを使用する，ビジョンベース＋モデルベースのアプローチにおける最新の研究では，「モデルの校正」,「手首姿勢と各指姿勢の同時推定」および「推定精度の定量的評価」，の三つの問題を解決することが特に重要な点となっている．本論文の目的は，これら3つの問題をユーザ固有の手モデルの生成と予測フィルタを用いた推定手法によって解決することである．

推定に利用する手モデルの表面形状表現方法の改善及び，リンク長などのモデルパラメータを個々のユーザに対応することは推定精度の向上につながる．しかし従来研究では，MRIのような高価な装置を用いて調整するものや，姿勢推定を行いながらモデルパラメータの調節を行うものが多く，簡単に個人別の手モデルを作成することが困難である．そこで，性別や手の大きさなどの物理的な違いに関係なく，誰もが簡単に手形状推定システムを利用できるように，推定システムと同じ複数のカメラを使用した手モデルの個人対応手法を提案する．論文前半では，姿勢推定する前に行う手モデルの校正手法について述べる．提案手法では，手の表面形状を視体積交差法を用いてボクセル表現する．さらに校正のための手指動作を観測することで，手モデルと観測結果の誤差を最小化し，各指のリンク長を推定する．実際の各関節間の長さと推定したリンク長を比較することで，提案手法の有効性を確認した．さらに，手の姿勢推定の性能を定量的に評価するた

[*] 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文, NAIST-IS-DD0661027, 2010年9月3日.

め，個人対応していない手モデルと個人対応した手モデルによる姿勢推定結果の比較を行った．

手の姿勢推定に関する多くの研究では，手首位置姿勢を設定するグローバルパラメータ，あるいは指関節角度を設定するローカルパラメータのいずれかを固定して手姿勢を推定している．しかし，このような制限は，ユーザにとっては自然な手の動きを困難にするものとなる．これら両方のパラメータを同時に推定するために，本研究では，非線形フィルタの一種であるアンセンテッドカルマンフィルタ（UKF）を手形状推定に適用することを提案する．アンセンテッドカルマンフィルタを用いれば，手モデルと観測の誤差を最小にすることが可能となる．15自由度を持つ手動作推定を通して，手首位置姿勢と指姿勢の同時推定実現可能性を検証した．

本論文では，手モデルの校正と予測フィルタの導入により，現在の手姿勢推定システムの主要な問題に取り組み，その結果，手モデルの個人対応により姿勢推定精度の改善を達成し，さらにアンセンテッドカルマンフィルタの適用によって，自然な手動作における姿勢推定を可能にする，グローバルとローカルパラメータの同時推定を実現した．

キーワード

vision-based hand pose estimation, hand model calibration, Unscented Kalman Filter, multiple viewpoint cameras

Contents

List of Figures

List of Tables

List of Abbreviations

CM	Carpometacarpal
DIP	Distal Interphalangeal
DOF	Degree of Freedom
DOFs	Degrees of Freedom
EKF	Extended Kalman Filter
GA	Genetic Algorithm
HRI	Human-Robot Interaction
HCI	Human-Computer Interaction
ICP	Iterative Closest Point
KF	Kalman Filter
MCP	Metacarpophalangeal
PCA	Principal Component Analysis
PF	Particle Filter
PIP	Proximal Interphalangeal
SA	Simulated Annealing
SMA	Specialized Mapping Architecture
UKF	Unscented Kalman Filter

Chapter 1

Introduction

> Where the spirit does not work
> with the hand, there is no art.
>
> Leonardo da Vinci

1.1. Background and Motivation

The importance of the hand in our daily lives is best shown by the amount of muscular control our brain devotes to it. Penfield and Rasmussen [1] in studying cerebral cortical functions, discovered that almost a quarter of the brain's motor cortex is devoted to moving the hand. This discovery is illustrated by the motor homunculus in Figure 1.1, which also shows the body organs' sizes as a function of the amount of brain's motor cortex associated with them. The sizable brain power devoted to the hand endows it with the complexity, dexterity, and sensitivity that have enabled human beings to achieve technological breakthroughs that greatly improve quality of life. Using his hands, man was able to create tool from the simplest, like axes, to the most complex, like computers. It is, in fact, the same hands that had ushered the era of computers and robotics as we know it.

As we become more dependent on computers and robots in our daily lives, from browsing web pages to operation of complex machineries, the need to make the interaction between users and machines more intuitive also broadens. The field of Human-Computer Interaction (HCI), as illustrated in Figure 1.2, encom-

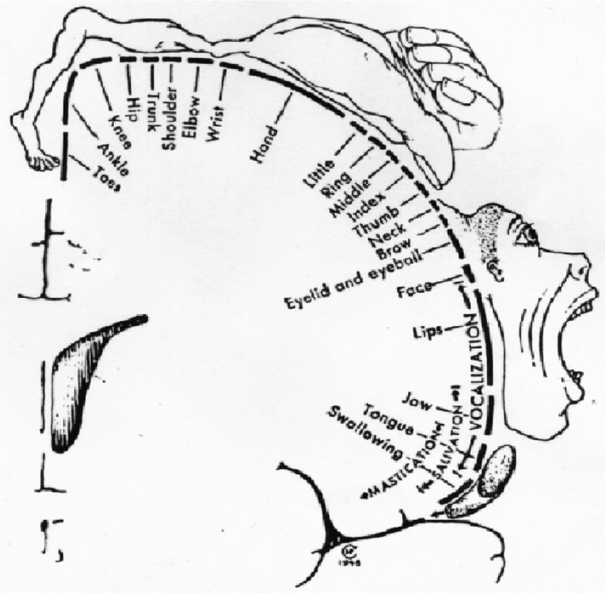


Figure 1.1: Motor homunculus, adapted from [1]. Take note of the size of the hand compared to the other organs. The sizes of the body organs are illustrated as a function of the amount of brain's motor cortex associated with them.

passes this desire to improve the integration of automated systems into our lives. While the left side of Figure 1.2 shows the disciplines needed to understand and model the dynamics of human interaction, the right side shows the component of computer systems involved in an HCI. An important aspect of improving HCI is enabling the interaction between the user and the computer or robot to be as easy, as natural, and as convenient as possible.

For bulky systems like computers and robots, mouse, keyboard, and joystick are the most prevalent input tools for robotic systems. These tools tend to limit the form of interactivity between the machine and the user. For industrial type robots like those used in manufacturing, keyboard or joystick might be the appropriate interactivity tool. Unfortunately, different types of robots or activities require different input tools. More anthropomorphic forms of robots, like humanoid robots, or activities like robot-aided surgery, necessitates a rethinking of the input tools. Using mouse or keyboard to do surgery aided by robot, for example, would severely limit the motion of surgeons. Development of new input tools for seamless interactivity between man and machine is necessary.

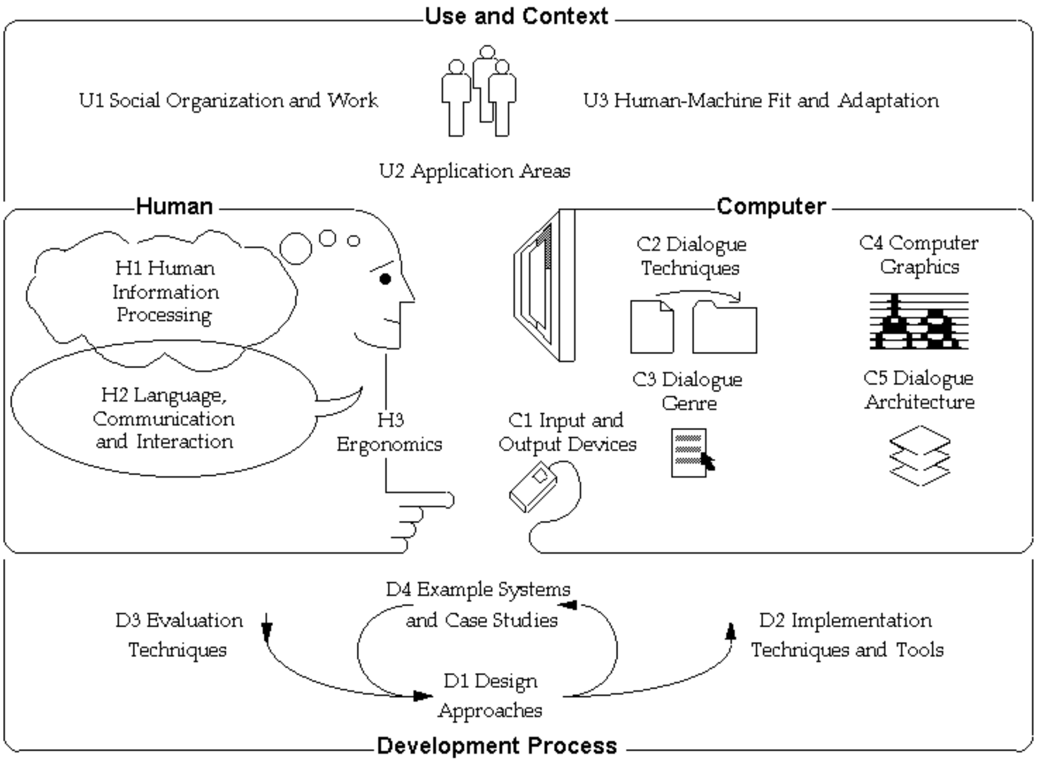

Use and Context
U1 Social Organization and Work
U3 Human-Machine Fit and Adaptation
U2 Application Areas
Human
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
Computer
C2 Dialogue Techniques
C4 Computer Graphics
C3 Dialogue Genre
C5 Dialogue Architecture
C1 Input and Output Devices
D4 Example Systems and Case Studies
D3 Evaluation Techniques
D2 Implementation Techniques and Tools
D1 Design Approaches
Development Process

Figure 1.2: Overview of Human Computer Interaction [2].

The rigidity and inflexibility of traditional input methods like mouse or keyboard relate how the hand is used, i.e., as an intermediary in the transmission of information rather than as source of information itself. In using the mouse for example, clicking a button transmits a particular message that is interpreted by the computer system, like when selecting "cancel" on a webpage. Intuitively, the same information can be conveyed directly by the hand itself by making a "stop" sign (open hand and palm with the fingers fully extended). In short, hand gestures are the key to a more natural interaction between man and machine pointed out by Pavlovic *et al.* [3].

Vision-based systems, which use camera as their input device, do not need bulky measuring devices like datagloves to determine hand gesture. Various applications developed by different researchers have demonstrated the desirability of robust vision-based gesture interpretation systems. These applications vary including finger painting [4], computer game controller [5], and robot controller

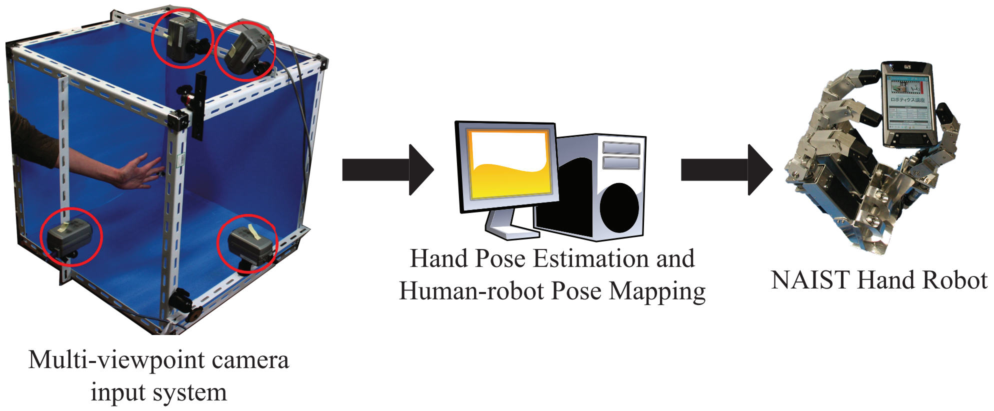


Figure 1.3: An example application: a system for robot hand control using hand pose. The captured hand pose by the multi-viewpoint camera system can be used to directly control a robot hand by mapping the pose to the robot hand's kinematic structure.

[6].

An important step in recognizing hand gestures, both communicative and manipulative, is determination of the hand pose. Vision-based hand gesture or pose recognition would help in achieving an easy and natural interaction between man and machine, characteristics desired for HCI and HRI systems. Such a system would enable the users to interact seamlessly with computer or robotic systems. In the example illustrated in Figure 1.3, the hand pose captured by a multiple camera system can be mapped to a robot hand's kinematic structure, thus enabling direct robot control by hand motion.

1.2. Research Aim and Approach

Three important issues in hand pose estimation are addressed by this thesis: model calibration, quantitative evaluation, and flexibility. These issues are not addressed fully even in state-of-the-art research of hand pose estimation. This thesis hopes to contribute to the development of a vision-based hand pose estimation system that allows unrestricted hand motion as input.

The approach proposed by this thesis to address the three problems is two-pronged: calibrate the hand model for every user and apply predictive filtering. First, the hand model used in pose estimation is calibrated to match the actual shape of the user's hand. This can improve pose accuracy by minimizing disparities between the system's model and the actual hand. Moreover, using the calibrated hand model allows quantitative comparison with another pose measurement method, the dataglove. Second, Unscented Kalman Filter (UKF) is used to track the hand motion and estimate it's pose at any given time. The time series information obtainable from a moving hand improves estimation accuracy and error recovery by allowing the system to predict the possible path of the hand's motion. Using UKF has the added benefit of simultaneously estimating global and local parameters of the hand. The global parameters are the wrist position and palm orientation while the local parameters are the finger joint angles.

The contributions of this work to hand pose estimation research are the following:

- A model calibration method that enables a hand pose estimation system which accommodates different users regardless of age, gender or physical hand characteristics and kinematics;
- A quantitative evaluation of a vision-based hand pose estimation system;
- Simultaneous estimation of global and local hand pose parameters, allowing greater flexibility for the user.

1.3. Thesis Layout

The remainder of this thesis is organized into the following parts. Chapter 2 covers the current state-of-the-art in model-based hand pose estimation, applications of gesture and pose recognition in HRI and the different hand models used in various hand pose estimation systems. Chapter 3 discusses the approach implemented to calibrate the hand model, including its experimental results. Chapter 4 presents the implementation of the Unscented Kalman Filter in the hand pose estimation system. Implementation details as well as simulation results are discussed. Lastly, Chapter 5 outlines the conclusions from the previous chapters and summarizes the thesis. The Appendix contains supplementary materials such as UKF details, the voxelization process, and additional motion and hand data, among others.

Chapter 2

Literature Review

> Only those who will risk going too far can possibly find out how far one can go.
>
> T. S. Eliot

The hand is a very articulate and versatile tool for communication and manipulation. Being able to capture the motion of the hand would allow a computer or robot system to understand what the user want to do or express. A lot of research has been done toward capturing hand motion and interpreting hand gestures. For this research, the application of vision-based techniques, since it allows natural hand motion as input, is emphasized. Thus, most of the papers which will be discussed in this chapter, and in the rest of this thesis, are vision-based, i.e., camera is used to track the hand motion or pose.

This chapter presents an overview of related works in order to show where this thesis lies in the field of hand pose estimation. The first section discusses relevant model-based techniques, as opposed to appearance-based, of estimating the hand pose. Then the next section focuses on complete hand pose estimation works followed by an introduction of the use of hand motion tracking in human-robot interaction. The chapter is concluded by a special section devoted to the different hand models used in hand pose estimation research.

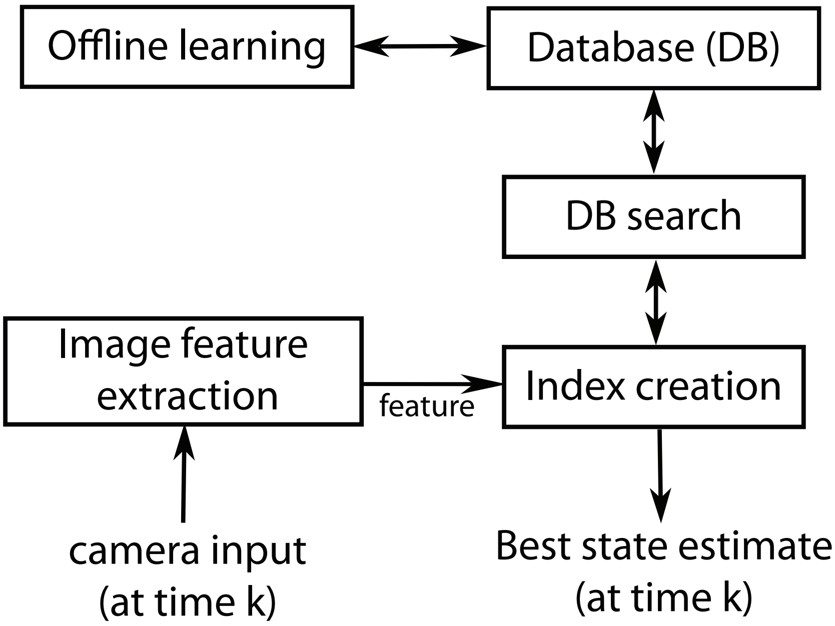


Figure 2.1: Outline of an appearance-based system.

2.1. Hand Pose Estimation Approach

Appearance-based approach relies on the hand image as the input data. Figure 2.1 illustrates the pose estimation process of an appearance-based approach. First, geometrical or statistical information of the image, also called feature set, is extracted to create an index that corresponds directly to a known hand pose configuration. Then, nonlinear mapping of indexes to hand pose configurations is learned off-line using a large set of image data. During operation at time k with the given feature set obtained from an input image, the database is searched for a corresponding hand pose. The returned hand pose is the pose estimate at time k.

Segen and Kumar [7] provide an example of an appearance-based approach. In their paper, they determined the 3D position of the hand using shadow. They were also able to determine if the hand pose belongs into one of four categories: point, reach, click, and ground. The features they used include the outline of the hand silhouette, and the number of its peaks and valleys. Since there is a small number of distinct gestures to categorize, they did not need to employ a database of indexes.

In the paper of Shimada *et al.* [8], the hand features used to create the indexes for the database were fingertips, finger axes and palm contours. To compress the database and speed up comparison, similar features were grouped using Principal Component Analysis (PCA). Grouping had the added benefit of ensuring that every hand pose has a match (or near-match) in the database. During operation, the features of the input images are converted into an index which is then compared to the indexes available in the database.

Athitsos *et al.* [9] also built a database of known hand configurations, however, they made an improvement over the work of Shimada *et al.* by allowing the hand pose to be determined even in a cluttered environment. They solved the cluttering problem by doing two things. First, the binary edge images were embedded in a high-dimensional Euclidean space which was used to approximate image-to-model chamfer distance efficiently. Second, the probabilistic line matching was ran which identifies line segment correspondences between the model and the input image that are the least likely to have occurred by chance.

Other works that essentially follow the pattern outlined in Figure 2.1 include [10, 11, 12, 13, 14]. Appearance-based approach has the advantages of speed and efficiency in terms of calculation and algorithm, especially if the mapping is learned off line. However, the learning process itself takes a lot of time, especially when working with a large database of hand pose. Unfortunately, a large database of images does not always guarantee a high recognition rate [9]. Appearance-based approach is more suited to applications that are very specific about the context and the variety of required hand pose. For example in [7], the system requires the use of visible light since they used shadow to calculate the hand pose, limiting its potential applications. In a review conducted by Pavlovic *et al.* , appearance-based approach is perspective-limited and usually gives solution only to a specific task problem.

An alternative, the model-based approach, allows a system to estimate the hand pose without being tied to any particular application or camera perspective. This approach employs parametric modeling of the human hand, wherein certain parameters describe the behavior and characteristics of the hand model. Model-based method tries to minimize the error between a predefined model of the hand

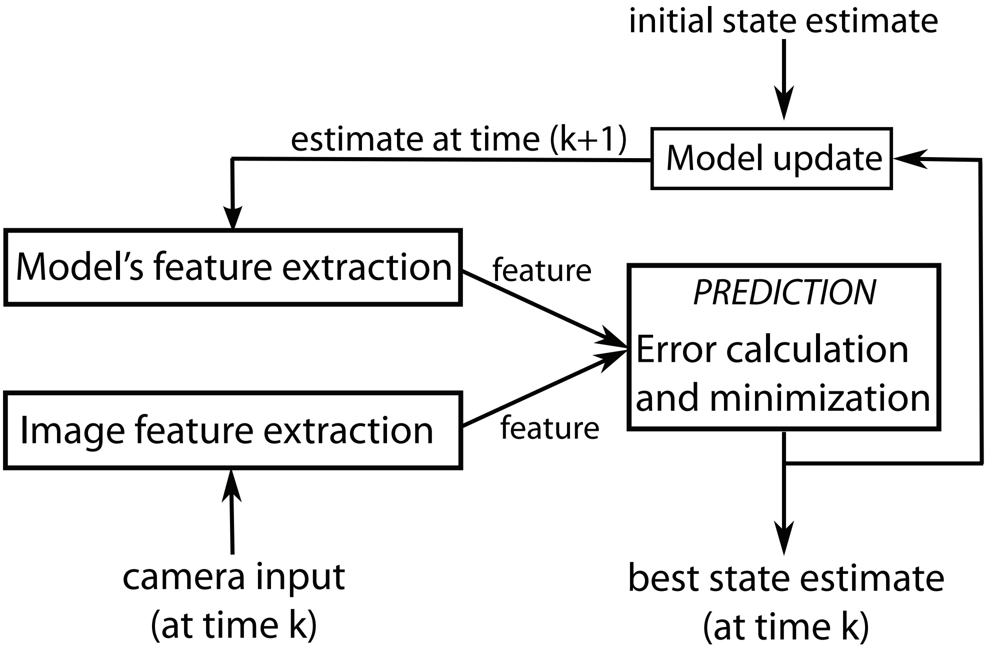


Figure 2.2: Outline of a model-based system.

and the observation data, in order to generate precise results that can be used by any system. Some of the model-based approach research works that stand-out are [15], [16], [17], [18], [19], [20], [13], [21], and [22].

Figure 2.2 illustrates a model-based approach. First, at a given time k, features from the input image and the model are extracted and compared. Then the error is minimized in order to determine the best state of the hand pose at time k. This step is also known as analysis by synthesis because the model pose is refined according to its matching with the input image. The error calculation and minimization step is also called the prediction step because it uses information from both the current data (camera input image) and the previous data (model's state estimate) to obtain the best possible state estimate at time k. The best estimate then becomes the model's state at time $k + 1$.

An overview of the different papers discussed in the rest of this section is presented in Table 2.1. The table lists the research work along with the number of DOFs estimated by the implementation, the relevant features extracted, the estimation method, and the known limitations. The estimation method refers to

Table 2.1: Comparison of Selected Full-DOF Hand Pose Estimation Research

Researcher	**DOF Implemented**	**Feature Extracted**	**Estimation Method**	**Limitations**
Rehg and Kanade [15]	5 local; 3 global	Finger Tips	Least square	Occlusion, only 8DOF
Nirei *et al.* [17]	27 local; 6 global	Silhouette, Optical flow	GA with SA	Accuracy of optical flow
Heap and Hogg [18]	N/A	Edge	Weighted least squares	Limited DOF, Occlusion
Lien and Huang [19]	17 local; 6 global	Marker	GA	Palm must face camera
Delamarre and Faugeras [23]	21 local; 3 global	Silhouette and depth disparity	ICP	Palm must face camera
Lu *et al.* [21]	20 local; 6 global	Edge, Optical flow, Silhouette	Force model	Self occlusion
Stenger *et al.* [20]	1 local; 3 global	Edge	UKF	Limited DOF
Rosales *et al.* [13]	22 local; 2 global	Moment	2D-3D mapping using SMA	Initialization, hand-hand occlusion
Ueda *et al.* [22]	16 local; 3 global	Silhouette	Force inspired in 3D	Either local or global only
Stenger *et al.* [24]	N/A	oriented edge, color	Hierarchical Cascade of Filters	Initialization
Causo *et al.* [25]	16 local; 3 global	silhouette, voxel	Force inspired in 3D	Either local or global only
Causo *et al.* [26]	16 local; 3 global	silhouette, voxel	UKF	Slow computation

the optimization approach used to determine the best state estimate.

Rehg and Kanade [15, 16] did a pioneering work in model-based hand pose estimation. The system they developed estimates the hand pose in 3D in each frame and uses a hand model composed of cylinders, which mimics a simple kinematic structure. Extracted features reflect the underlying skeletal formation. These are lines and points obtained by detecting the occluding boundaries of finger links and tips from the input images. The same features are extracted from the model after projecting it through a camera model. Model fitting using non-linear least square method essentially finds the best hand configuration that fits the given state estimate. However, the system suffers from occlusion problems, including self-occlusion.

In [17], Nirei *et al.* represented the hand as a set of truncated elliptic corn that mimics the structure of the hand with 27 DOFs. Hand silhouettes of both the input image and the the hand model projection are the features compared. Overlapping of the silhouettes are maximized using a combination of genetic algorithm and simulated annealing. To help with the tracking of the fingers by ensuring a faster search for the best state estimate, optical flow is extracted. However, accuracy of the system is highly dependent on the accuracy of the optical flow extraction.

In Heap and Hogg's paper [18], edge features of a deformable hand model were used to track a 6 DOF hand. The system was trained, and used weighted least squares to estimate the pose parameters. It has the advantage of being usable with various regular sized hands because it has been trained on many samples. However, in addition to the limited DOF it can track, it also performs poorly when there is occlusion.

Lien and Huang [19] tracked a 23 DOF hand, using markers on the hand as features. They used genetic algorithm to determine the inverse kinematic solutions of the moving hand model. However, the palm of the hand must face the camera at all times during tracking in order for the system to work.

Delamarre and Faugeras [23] implemented a motion tracking system using a 27 DOF hand model and used multiple cameras to provide depth information to the input images. The system minimizes the disparity between the model and the input images using a force-driven iterative approach, similar to an Iterative

Closest Point estimation. The features used are the input image silhouettes and their depth disparity. However, the system suffers from finger self-occlusion and requires the palm to face the camera during operation.

Stenger *et al.* [20] used kinematic parameters as the basis of the quadric hand model. The feature derived from the input images is the silhouette edge, which is compared to the edge of the hand model's projection during the error-minimization stage. The quadric hand model is changed or moved according to the difference between the image features of the input and the model to reflect the actual pose of the hand. Their system used Unscented Kalman Filter to minimize the error between the model and the input image, and track the moving hand. However, it could track only 7 DOFs, 1 DOF for the local pose and 6 DOFs for the global pose.

A different approach was taken by Rosales *et al.* [13]. Instead of computing for the pose estimate online, they implemented a Specialized Mapping Architecture, a learning strategy that maps selected features to particular 3D hand pose. The system extracts moment-based features for comparison with the learned hand poses. They were able to estimate the pose of two hands, although they had problems with initialization and hand-hand occlusion.

Lu *et al.* [21] implemented a hand pose tracker which uses multiple features such as edge, optical flow and shading data. This is a novel approach in terms of integrating multiple cues to help track the hand movement. A forward recursive dynamic model is used to track the motion in response to the 3D data derived forces applied to the model. Although it can track a 26 DOF hand, it cannot handle self-occlusion due to finger movement while the hand is changing position or orientation.

Ueda *et al.* 's research [22] implemented a 31 DOF hand model, however only either of the global (25 DOFs) or the local (6 DOFs) pose was estimated at any given time. Silhouette images from multiple view-point cameras were used to create a voxel model of the hand. A surface model was fitted into the voxel model using physically-derived forces to estimate the hand pose.

Like Rosales *et al.* [13], Stenger *et al.* [24] also implemented a learning strategy to estimate the hand pose. The features used by the system for the learning step and during pose estimation are oriented edge and color and their derivatives, such as distance transform edge map. Additionally, their system required a fixed hand pose for initialization. The main difference between the two approaches was the kind of training image: Rosales *et al.* used synthetic data to train the system, while Stenger *et al.* used real images.

The last two entries in Table 2.1 describe my work and contextualize it with respect to other works in hand pose estimation. My approach is not without shortcomings just like the other state of the art works, but compared to them, mine has the advantage of enabling any user to use the system by calibrating the model. My approach also allowed quantitative evaluation of the pose estimation system. In using a predictive filter (Unscented Kalman Filter), error recovery improved and simultaneous estimation of global and local pose parameters enabled. The details of my approach are presented in the rest of this thesis.

For a more comprehensive review and discussion of appearance-based and model-based approaches to hand pose estimation and gesture recognition, Pavlovic *et al.* [3] and Erol *et al.* [27] are available.

2.2. Vision-based Hand Tracking for Human-Robot Interaction

This section presents works that further emphasize the applications of hand motion and gesture. Hand gesture recognition and hand motion tracking have been employed before to control robot systems without the use of tethered gadgets.

Waldherr *et al.* [28] controlled a robot cleaner that recognizes and tracks the human user and recognizes gesture. A combination of template matching and neural-network solution enabled the robot to recognize commands from the user. Although the recognized vocabulary is limited, it is enough to control a cleaning robot.

A service robot named ALBERT was developed by Rogalla *et al.* [29]. It is fully equipped with a laser range finder for navigation, speech recognition system for oral communication, vision system for gesture detection and object recognition, and robot arm and hand for object grasping and manipulation. The robot recognizes up to six gestures which can trigger events that have been preloaded in its database. For example, it can detect a pointing gesture including the object being pointed and upon recognition of the object loads and executes a series of events, like picking-up the object. The robot is an example of a system that learns by *programming by demonstration.*

Stiefelhagen *et al.* [30] developed a multi-model interaction system for a robot. It has a speech synthesis and recognition and vision systems similar to [29]. The speech recognition system implements a dialogue component while the vision system can recognize pointing gestures and head orientation. For the gesture recognition, the hand is detected through its color, like the hand recognition system in [29].

Kofman *et al.* [31] envisions a system wherein the user is unencumbered by wired gadgets. They implemented a vision-based robot manipulator teleoperator with six DOFs. The system can remotely control a six-axis industrial robot arm in order to place objects on a target. The system is fully vision-based, and the hand is tracked using markers and two cameras.

Interaction with a robot is not limited to communicative level but also manipulative. The hand gesture or pose made by the user doesn't have to be interpreted by a robotic system as having some other meaning. For example, Infantino *et al.* [32] built a system that directly controls a hand robot by mimicking the shape of the user's hand. Instead of interpreting the pose of the user hand, the robot simply mimics it. Their work used calibrated stereo cameras and a novel visual servoing technique to quickly locate the fingertips and estimate the hand pose. They demonstrated the robustness of their system with the robot's more than 70% success rate in mimicking 17 different hand poses.

On a related work, Infantino *et al.* [33] demonstrated a teaching system for anthropomorphic hand robot using vision-based system, the same system in their earlier work [32]. To teach the robot, they implemented a cognitive approach which maps the hand pose to primitives in situation and action space. With their technique, they achieved a reasonable success rate in playing rock-paper-scissor game.

These are just some of the works that incorporate tracking of the human hand using vision-based systems in robotic applications. The robotic platforms used vary from industrial type to humanoid type. Although the discussion is outside the scope of this thesis, these two papers could be of interest to the readers: Thrun

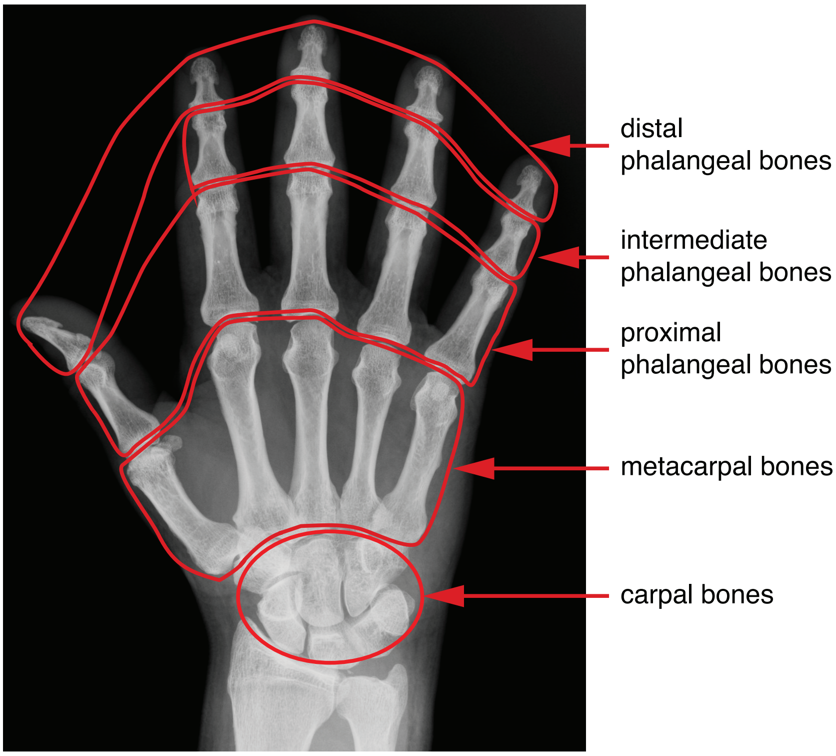


Figure 2.3: The hand skeleton showing the different hand bones [36]. Take note of the absence of the intermediate bone in the thumb.

[34] and Goodrich and Schultz [35]. Both papers consider past works and current trend in HRI research and clear out the path future work can take, including the use of vision-based hand pose estimation system in HRI.

2.3. Hand Model

The hand is the primary organ of the body for manipulating the environment. Anthropologically speaking, the hand is the appendage at the end of each arm of primates. Human beings have two hands, while primates like apes and monkeys are considered to have four hands.

Each hand has four fingers and a thumb. The four fingers are known colloquially as the index or forefinger, the middle finger, the ring finger, and the little finger or the pinkie. The medical name for each one is digitus secundus manus,

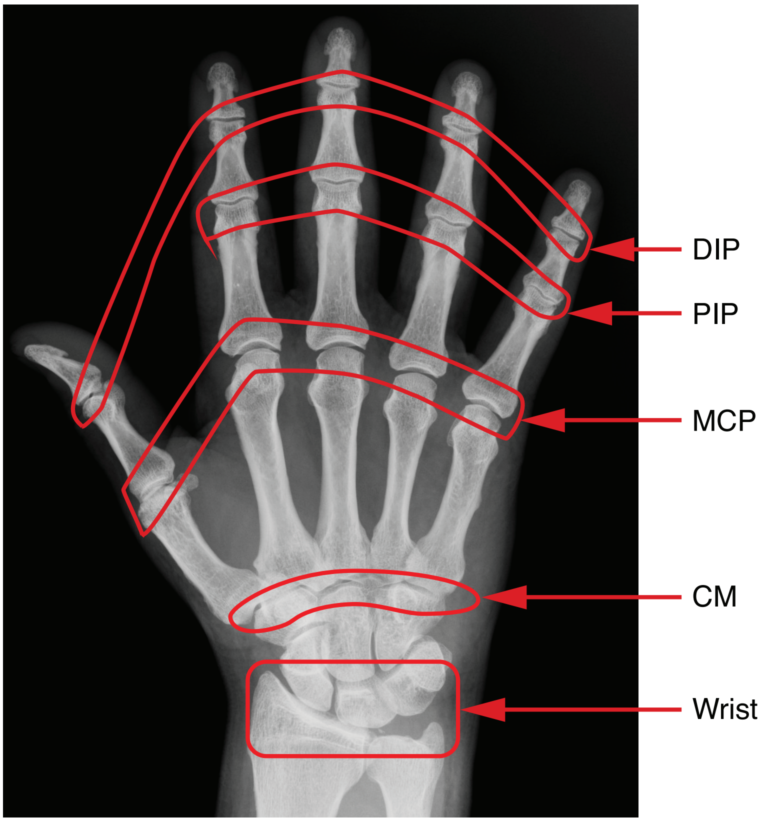


Figure 2.4: The finger joints. Take note of the absence of PIP joint in the thumb.

digitus tertius, digitus annularis, and digitus minimus manus, respectively. The thumb, the first digit of the hand, is known as the digitus primus or pollex in Latin. What separates primates, humans most especially, from other animals that possess hand-like appendages is the presence of the opposable thumb.

As shown in Figure 2.3, the human hand has 27 bones: eight is located in the wrist area, five in the palm area, and the rest are in the fingers and thumb. There are two rows of bones in the wrist or carpus. From the wrist, five bones or metacarpals extend to the palm area. Extending from the metacarpals, are the finger bones also known as phalanges or phalanx bones. The first row of phalanges is called the proximal phalanges, the next row is the intermediate phalanges, and the last row is the distal phalanges, which make up the finger tips. The four fingers have three rows of phalanges, while the thumb possess only proximal and distal phalanges [36].

Table 2.2: Hand Joint Definition

JOINT	CONNECTED BONES
Carpometacarpal (CM)	Carpal and Metacarpal
Metacarpophalangeal (MCP)	Metacarpal and Proximal Phalange
Proximal Interphalangeal (PIP)	Proximal and Intermediate Phalanges
Distal Interphalangeal (DIP)	Intermediate and Distal Phalanges

The hand's articulation is defined by the bones it connect. For the rest of this thesis, a naming system for the relevant hand joints are tabulated in Table 2.2 and illustrated in Figure 2.4. Each finger possess CM, MCP, PIP, and DIP. The thumb, however, has only three: CM, MCP, and DIP.

The CM and MCP joints have two DOFs each, defined by two action pairs. The first pair is the extension-flexion and the second is the adduction-abduction. As shown in Figure 2.5, flexion for the MCP joints involves movement of the finger toward the palm; extension is toward the opposite direction. Adduction involves the movement of the fingers toward (coming together) the finger of the joint of interest; abduction is the movement in the opposite direction (coming apart). While the CM and MCP have two DOFs, the PIP and DIP have one each. Thus, every finger has a total of six DOFs.

The thumb's DIP and MCP have the same number of DOF as the fingers, but not its CM. The thumb's CM is usually considered to have two DOFs even though it has two non-orthogonal and non-intersecting rotation [37]. This is usually modeled as having two DOFs, but to accommodate the flexibility of the thumb, sometimes a three DOF CM is used [38]. Thus, the thumb can have five or six DOFs. In the model used in this thesis, the thumb has five DOFs.

The wrist has six DOFs: three for its location in 3D (x, y, z) and three for its orientation (roll, pitch, yaw); the orientation is actually for the palm. With the inclusion of the wrist, the total number of DOF of the hand is 35 (or 36, if the thumb's CM is modeled as a three DOF joint). For clarity in discussion, the wrist's DOF will be referred to as global pose parameters while the fingers' and the thumb's as local pose parameters.

Models used to represent the hand try to capture its kinematic characteristics. A full DOF hand has at least 31 parameters composed of 6 global and 25 local.

Although for the sake of faster calculation, the DOF is sometimes minimized. To lessen the number of parameters, the palm is modeled as a rigid part and the CM is fixed to zero DOF. This reduces the number of DOF of the local pose parameters to 21. The total DOF for the hand becomes 27.

Aside from fixing some joints to have zero DOF, constraints can also be applied. A very common constraint gives the value of the DIP joint angle relative to the PIP as shown in Equation 2.1. This same constraint will be used in the experiments for this thesis.

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP}. \tag{2.1}$$

There are three main groupings for the hand models: geometric, statistical, and physical-based. The hand is usually modeled geometrically by using various geometric primitives to represent the physical structure of the hand. Stenger *et al.* [20] used truncated quadrics to represent the hand, while Wu *et al.* [14] used a flat cardboard. Sometimes, this limits how the hand must be viewed, for example in [14], the palm must always face the camera orthogonally so the model could be fitted properly.

Due to the high number of DOFs of the hand and the multitude of poses possible, some works resort to the use of statistical hand model. Instead of capturing the skeletal, kinematic or physical characteristics of the hand, statistical characteristics of the hand pose is derived by sampling a lot of pose data. Heap and Hogg [18] applied Principal Component Analysis (PCA) on MRI data of the hand to model the different poses. Lin *et al.* [39] also used PCA, but the data is from joint angle measured by a data glove. They managed to extract a seven dimensional space to describe the different hand poses. Other works that used pose data from data glove to generate synthetic hand poses include [8], [40], [13], and [41].

Lastly, the physical-based model tries to emulate the actual hand as much as possible. Moreover, this kind of model takes into account the effect of various forces on the hand pose. More importantly, the skeletal structures of these models usually have a skin covering that deforms according to the shape of the underlying structure, thus, mimicking actual human hand shape change. Kuch *et al.* [38] created a skeletal hand model and covered it with a skin made of B-spline surface.

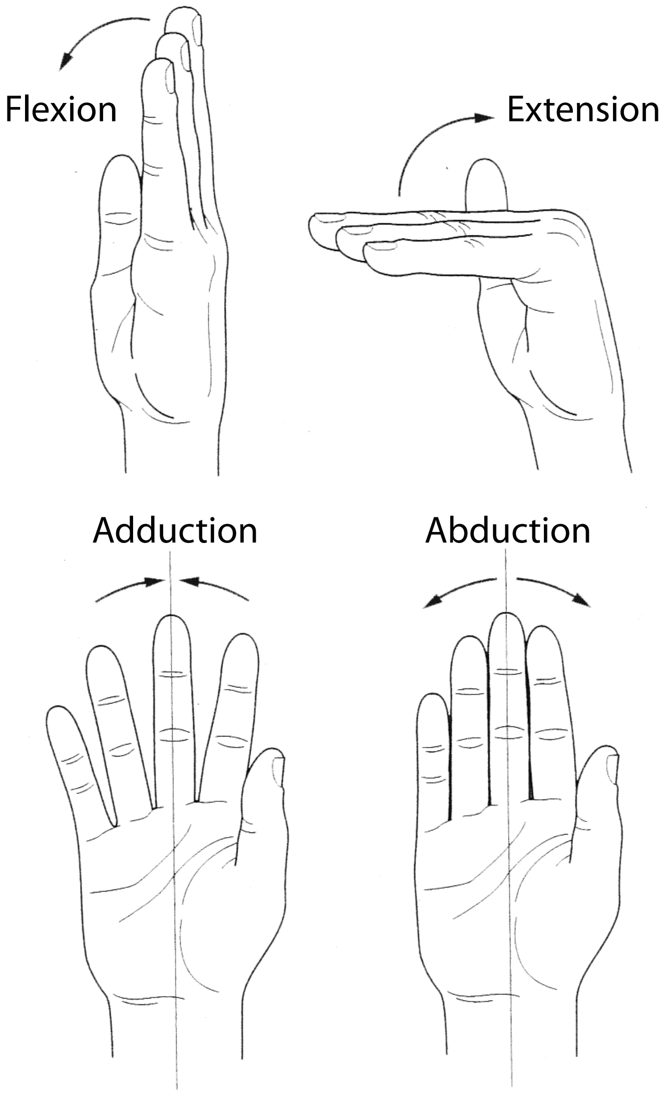


Figure 2.5: Motions of the MCP joint: extension and flexion, adduction and abduction [36].

Ueda *et al.* [22] also used a skeletal hand model with a quadric surface. Bray *et al.* [42] tracked hand motion using a model with polygonal skin and an underlying skeletal structure. Causo *et al.* [26] used a similar skeletal model as in [22] but the skin is made of voxels instead.

2.4. Conclusion

In this chapter, the state of the art in vision-based model-based hand pose estimation were reviewed. This establishes the ground where this thesis is based on. Additionally, works in the field of human-robot interaction that use hand gesture tracking or hand pose estimation were also discussed. This emphasizes the importance of this thesis in its application in making interaction between humans and robots and computers more natural and robust. Lastly, the anatomical characteristics of the hand is presented along with a short examination of the different models used to represent the hand. A key factor in hand pose estimation is the system's hand model accuracy. The choice of model to use in hand pose estimation depends on the pose estimation technique, with special consideration to the estimation speed and the desired result's accuracy. This thesis will show that employing user-adapted model can improve pose estimation.

Chapter 3

Hand Model Calibration

> The wise adapt themselves to
> circumstances, as water moulds
> itself to the pitcher.
>
> Chinese Proverb

3.1. Introduction

Problem Statement To enable any user to use a hand pose estimation system regardless of gender or physical differences such as hand size, hand model individualization using only multiple cameras is proposed. From the calibration motion, the proposed method estimates the finger link lengths as well as the hand shape by minimizing the gap between the hand model and the observation. We confirmed the feasibility of our proposal by comparing 1) calibrated link lengths and manually measured ones; and 2) hand pose estimation results using our calibrated hand model, a prior hand model and data obtained from dataglove measurements.

3.1.1 Related Works on Hand Model Calibration

Calibrating a hand model to fit a particular user is a difficult task, simply because there are too many parameters to take into consideration. So far, there are few works that explicitly calibrates the hand model prior to pose estimation. The usual way is an indirect approach to calibrating the hand model by estimating the parameters like link length and joint position simultaneously with the pose parameters. Shimada *et al.* [43] estimated the joint angles together with the widths and lengths of the fingers by applying constraints in order to narrow the search space to the candidate hand pose. Thayananthan *et al.* [44] also applied contraints in the form of chamfer distance and shape contexts to estimate the pose of the hand together with the width and length of the fingers.

The earliest study on the relation between the physical characteristics of the hand and its kinematics was done by Buchholz *et al.* [45]. They collected measurement data from hundreds of subjects in order to collect anthropometric description of hand kinematics statistically. This is a study in ergonomics that was meant to benefit designers and other fields. For example, Chua *et al.* [46] employed constraints and finger length ratios taken from anthropomorphic studies like in [45]. Unfortunately, anthropomorphic data is not easy to incorporate when designing vision-based model-based hand pose estimation systems.

A more accurate approach to calibrate hand models would be the use of specialized machines like CT or MRI. Volume data from the machines are adjusted in order to derive an optimally accurate model. Kurihara *et al.* [47] used CT imaging to develop a realistic hand model which has an anatomically correct bone structure and deformable skin. They estimated the link structure from scans of the hand at various poses and constructed a polygonal mesh for the skin, resulting to a hand model that looks and moves realistically.

Rhee *et al.* [48] developed a hand model using MRI volume data to adjust the fully articulated hand model, which has a kinematic control that reflects the human skeletal structure and a skin that produces smooth deformation. The resulting hand models are kinematically correct and aesthetically pleasing. However, they are not practical in applications involving large number of users because they require time-, labor- and cost-intensive scanning machines.

In contrast, an approach that is cheaper, faster, and precise enough for tracking uses volumetric data from shape-from-silhouette techniques such as Szeliski's [49]. Hattori *et al.* [50] created full body models that is adaptable to users by adjusting an *a priori* uncalibrated model using surface normals from voxel data to refine the model. The calibration method proposed in this chapter, while con-

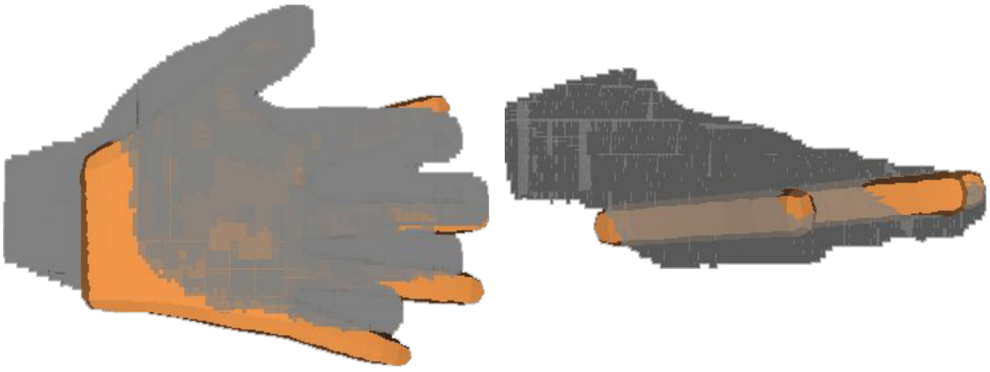


Figure 3.1: The observation data (gray hand) occupies a larger volume than the uncalibrated hand model (brown hand) used in [22].

ceptually similar to the work of Hattori *et al.* , assumes no prior model, except for the kinematic relationship of the finger links.

Model-observation mismatch could be minimized by taking advantage of the ability of multiple cameras and shape-from-silhouette approach to capture the shape of the hand. Figure 3.1 illustrates this disparity between the voxel data and the conventional hand model used in [22]; the uncalibrated model is smaller than the voxel data. The size difference makes the uncalibrated model fit inside the voxel data almost completely and less sensitive to shape changes due to the hand motion. To diminish model and voxel data shape disparity the hand model must be calibrated.

3.1.2 Proposed Method

In this chapter, a method for individualizing the hand model by calibrating its link lengths and shape is proposed. A multi-viewpoint camera system is used to calibrate the hand model and estimate the hand pose. This approach for calibrating the hand model does not require any special machine. Moreover the pose estimation is a single frame pose estimation technique [27].

The major contributions of this chapter are the model calibration method enabling adaptable hand pose estimation and the quantitative evaluation of vision-based hand pose estimation system. With calibration, the pose estimation system

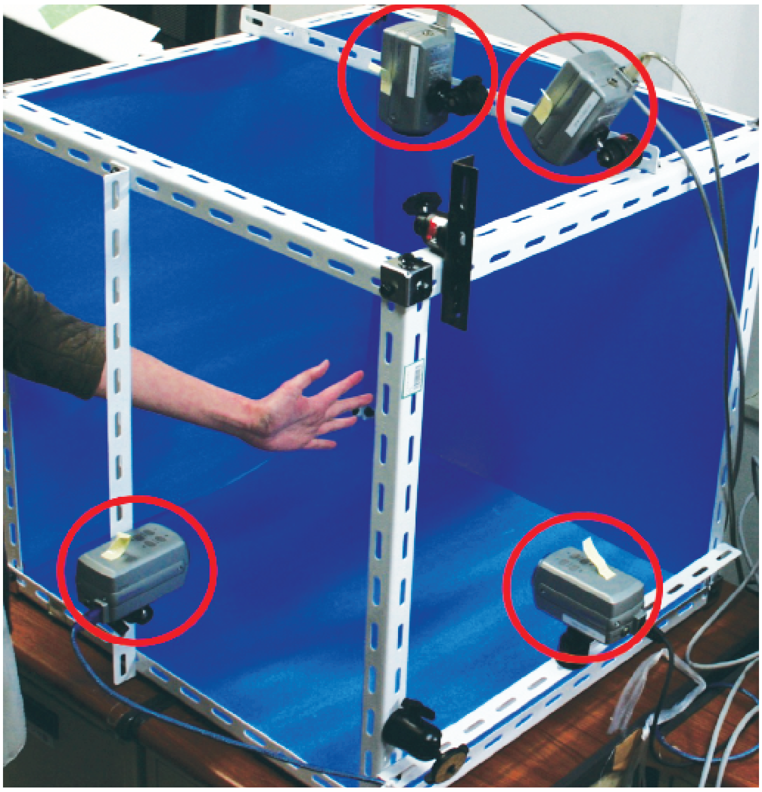


Figure 3.2: The pose estimation system using multi-viewpoint color cameras.

can be used by any user regardless of age, gender or physical hand characteristics and kinematics.

In contrast, dataglove requires calibration due to its fixed size and the fixed locations of the sensors on the dataglove. This raises problems with respect to hand size, shape, and fit. For example, a child's hand is too small for a dataglove. Figure 3.3 shows that when wearing a dataglove, similar shape and size profile results even for users with different hand characteristics. A vision-based system takes into consideration the disparities in user hand characteristics when estimating the hand pose.

This paper is also a first in vision-based hand motion tracking to present a quantitative performance comparison between two pose estimation techniques: the proposed method and a dataglove system. By creating a model of a hand wearing a dataglove, quantitative comparison of dataglove measurement and the proposed pose estimation is possible even in the absence of readily available

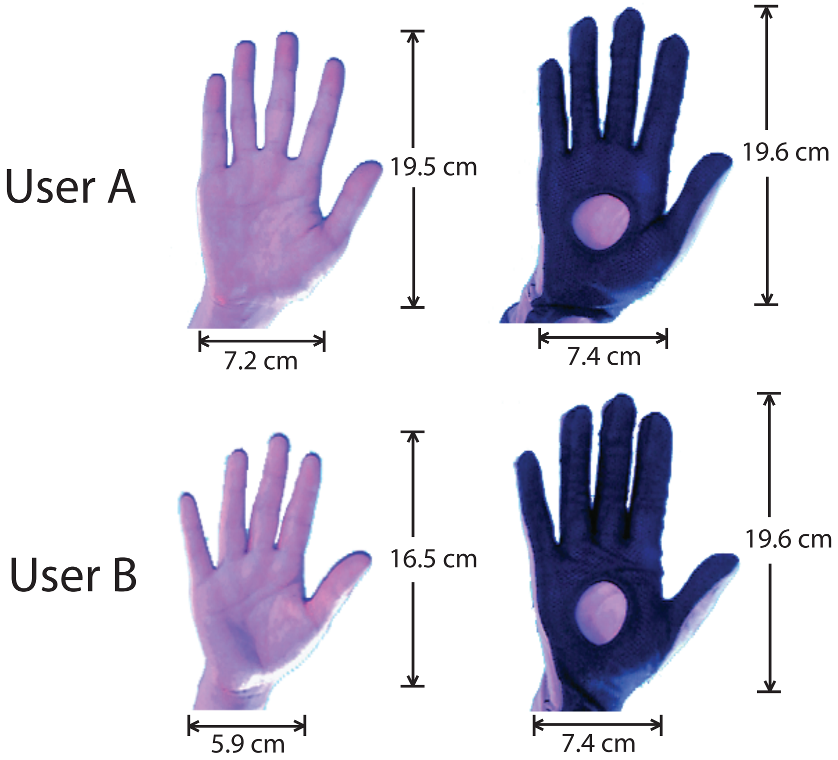


Figure 3.3: Dataglove use results in the same shape and size profile even for users with different hand shapes and sizes.

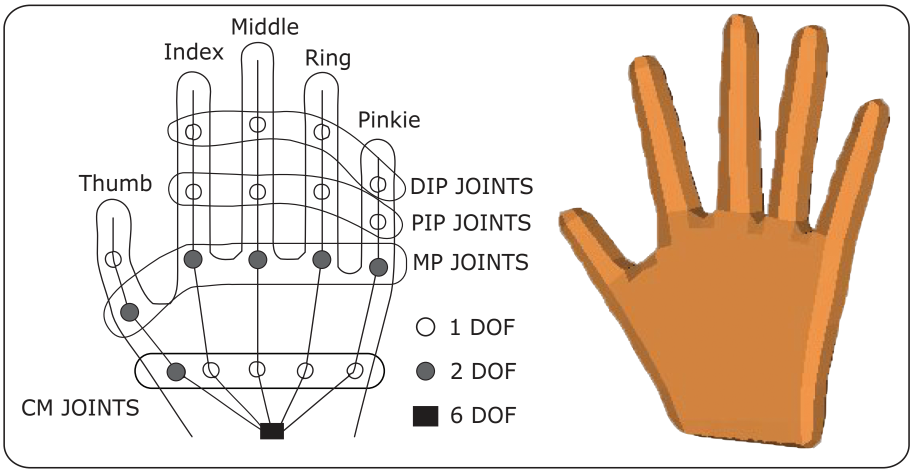


Figure 3.4: The uncalibrated hand model consists of a link structure representing the finger bones (left image) and a skin or surface structure made of quadrics (right image).

database of hand poses or motions.

3.2. The Hand Pose Estimation System

The pose estimation system, based on Ueda *et al.* [22], uses a hand model, hereafter referred to as the *uncalibrated model*, and multiple viewpoint cameras as shown in Figure 3.2. The uncalibrated model is a skeletal model which consists of a finger link structure, each of which is concatenated by a joint, and surface quadrics as shown in Figure 3.4. Camera images are converted to voxel data, which is used as observation data for the pose estimation. During pose estimation, the hand model is fitted to the voxel data by applying virtually generated forces as shown in Figure 3.6. The pose estimation process shown in Figure 3.5 is as follows:

1. Set the initial hand pose for tracking.
2. Get color input images from the multiple cameras.
3. Convert images to voxel using shape-from-silhouette technique [49].
4. Fit the model into the voxel data as shown in Figure 3.6.

   (a) Locate a part of the hand model that resides outside of the voxel data.

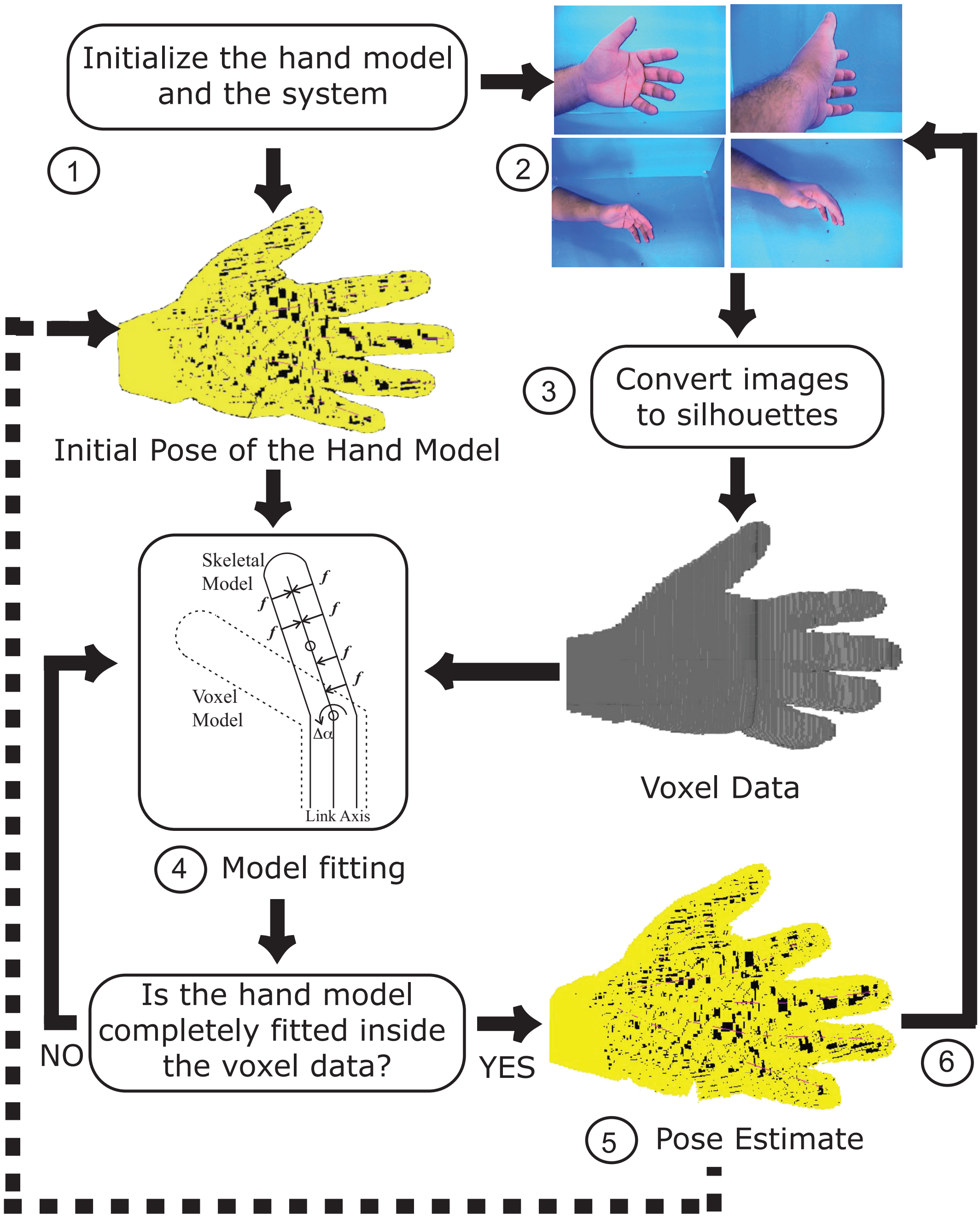

Initialize the hand model and the system
1
2
Initial Pose of the Hand Model
3
Convert images to silhouettes
Skeletal Model
Voxel Model
f
Δα
Link Axis
Voxel Data
4
Model fitting
Is the hand model completely fitted inside the voxel data?
NO
YES
6
5
Pose Estimate

Figure 3.5: The hand pose estimation process.

Table 3.1: Denavit-Hartenberg (DH) parameter of the hand model's links

Joint	θ	d	α	a
1	θ_1	0	$-\frac{\pi}{2}$	l_1
2	θ_2	0	$\frac{\pi}{2}$	0
3	θ_3	0	0	l_2
4	θ_4	0	0	l_3
5	θ_5	0	0	l_4

(b) Generate virtual force f on the exposed part.

(c) Push in the finger link associated with the exposed part by changing the joint angle by magnitude $\Delta\alpha$.

(d) Repeat from Step 4a until all exposed parts are inside the voxel data or when the fitting evaluation starts oscillating at a constant magnitude.

5. Update the hand pose.
6. Repeat from Step 2 for the next iteration.

3.3. CyberGlove User Study

Dataglove is the de facto uncalibrated when using hand motion as input to systems. The joint angles are detected by using gloves with attached strips that are sensitive to bending. The strips are located such that they are approximately over the finger joints when the glove is worn. However, such a configuration is also its shortcoming: different users, due to their varying hand sizes, would deform the strips differently. In other words, even for the same hand pose, the dataglove would have different readings for different users.

This section details the study done that shows that for different users doing the same hand pose, dataglove measurements also differ. Figure 3.7 shows the scanned hands of three different users. Each user hand is unique, but altogether demonstrates the variety in physical characteristics of the user hands. For example, User 1 has long fingers while User 2 and 3 do not. User 3 has thick fingers, while the other two have slender fingers compared to the over-all sizes of their

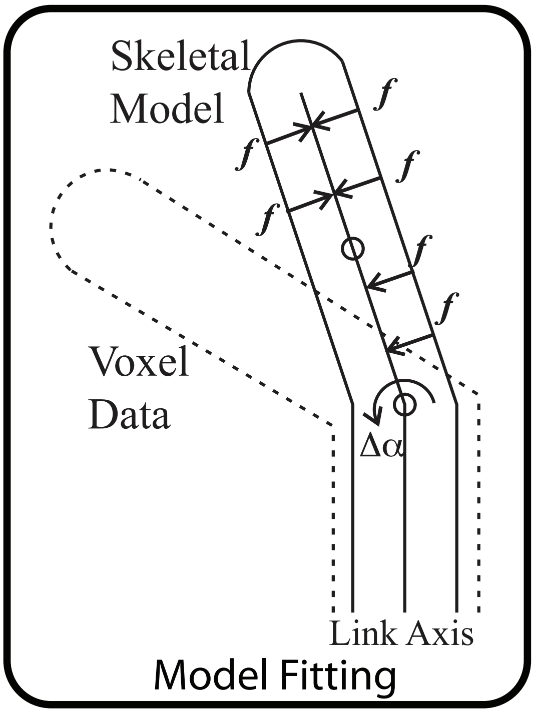


Figure 3.6: Model fitting generates virtual force f on the exposed parts of the skeletal model in order to push the model into the voxel data by $\Delta\alpha$.

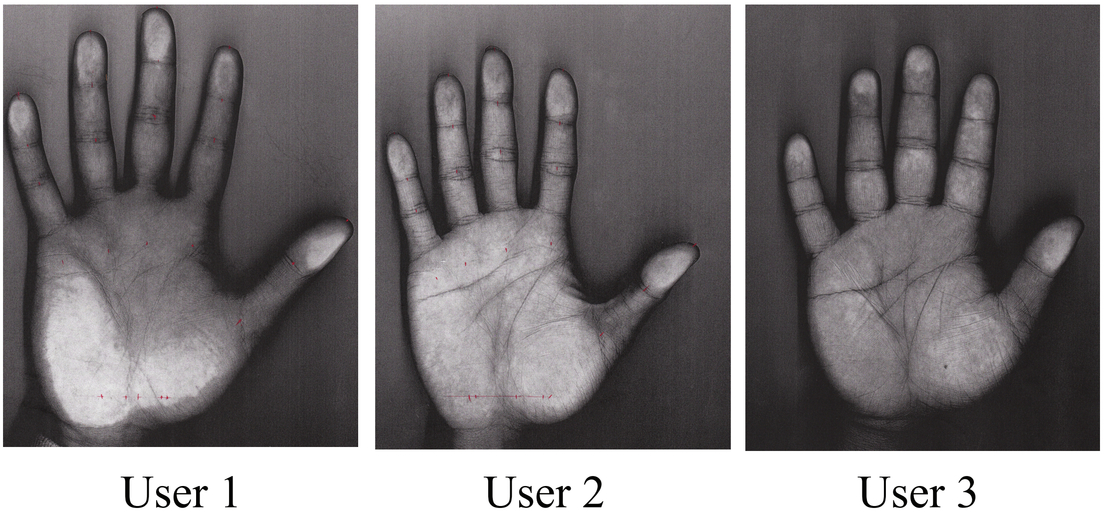
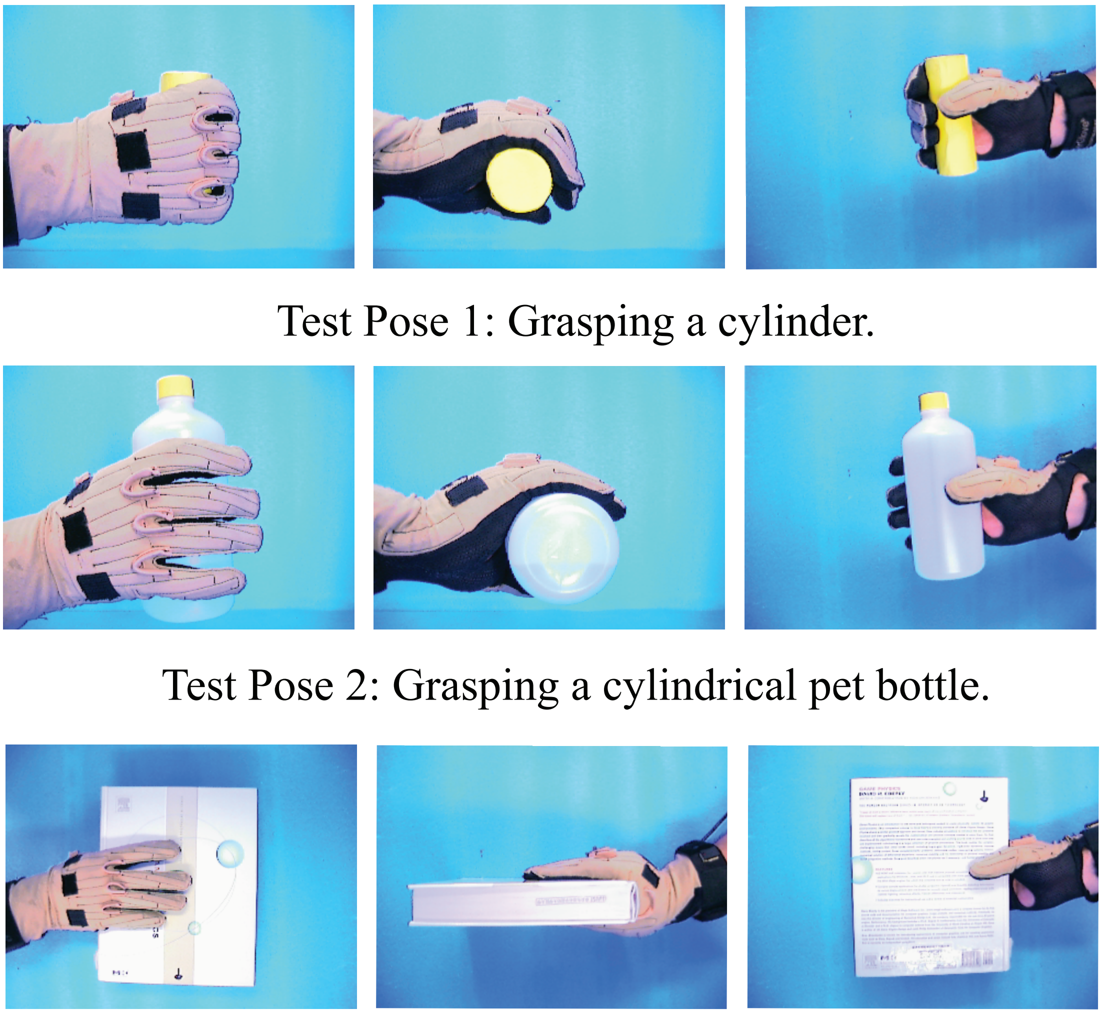


Figure 3.7: Scanned hands of three different user. The hands have different physical characteristics: the finger link lengths, the finger girth; and the over-all hand size differ from user to user.

hands. The link lengths of the three hands have been measured for comparison and listed in Appendix C.

The users made several pose by grasping objects with different sizes and shapes. The objects include a small paper cylinder, a plastic bottle, and a hardcover book. These were chosen to show different shapes of the hand. Grasping cylindrical objects requires all finger joints to bend. And since the paper cylinder and the plastic bottle have the same shape but different sizes, the effect of object size can be determined. The hardcover book was chosen because it required a hand pose whose PIP and DIP joints are supposed to be in the initial pose, i.e., flat. In Figure 3.8, grasping the yellow paper cylinder is labeled as Task Pose 1, grasping the plastic bottle as Task Pose 2, and grasping the book as Task Pose 3.

The dataglove measurements for the three task poses with three different users are presented in Figure 3.9, Figure 3.10, and Figure 3.11. The data presented are mainly for the Index finger's MCP joint since the measurement for other joints reveal similar patterns. In grasping the small paper cylinder (Figure 3.9), a difference of around 15 degrees can be noticed for the MCP joint. Surprisingly, the differences are smaller for the PIP joint but large for the DIP. Between User



Test Pose 1: Grasping a cylinder.

Test Pose 2: Grasping a cylindrical pet bottle.

Test Pose 3: Grasping a book at its spine.

Figure 3.8: A hand wearing dataglove while grasping objects of different sizes and shapes. The objects are a small paper cylinder, a plastic bottle, and a hardcover book.

1 and User 2, the DIP measurement differs by as much as 10 degrees.

When grasping a cylindrical object with larger diameter (Figure 3.10), the joint angle values increased proportionally. However, between the three users, the disparity between User 2 and User 3 have diminished considerably. A quick look at Table C.1 shows that the link lengths of both users are much nearer compared to User 1's. For a large hand, the dataglove gives a higher measurement value than for smaller hands.

When grasping a hardcover book, the DIP and PIP joints are in position referred to in this thesis as *initial pose* or in a flat position while the MCP joint is bent. In Figure 3.11, the difference in the MCP joint measurement is not so large. The fluctuation in the MCP measurement is probably due the hand readjusting its grasp since the book is a little heavy. However, for the PIP and the DIP joints, a constant difference can be observed between the three users. Between User 1 and User 2, there is a difference of around 5 degrees, and a 10 degree difference between User 2 and User 3.

To summarize, regardless of the size and shape of the object, and the hand pose, the dataglove gives different readings for different users. Even when the hand is at an initial pose, i.e., the fingers are in a flat pose, there is a clear difference in measurement values. This is not a problem if the dataglove will be used in a system by only one user to which the dataglove has been calibrated for. Unfortunately, for a system to truly allow natural hand motion, it must be able to accommodate different users.

3.4. Hand Model Individualization

In considering model accuracy with respect to the uniqueness of each user hand and the observation data needed by the system, an accurate hand model should reflect the input data correctly otherwise, pose estimation results would be ambiguous. Individual user hand shape and size must be accounted for.

3.4.1 Hand Model Structure

Our proposed hand model, a skeletal model, consists of a link structure representing the skeleton and a surface structure representing the skin and the hand

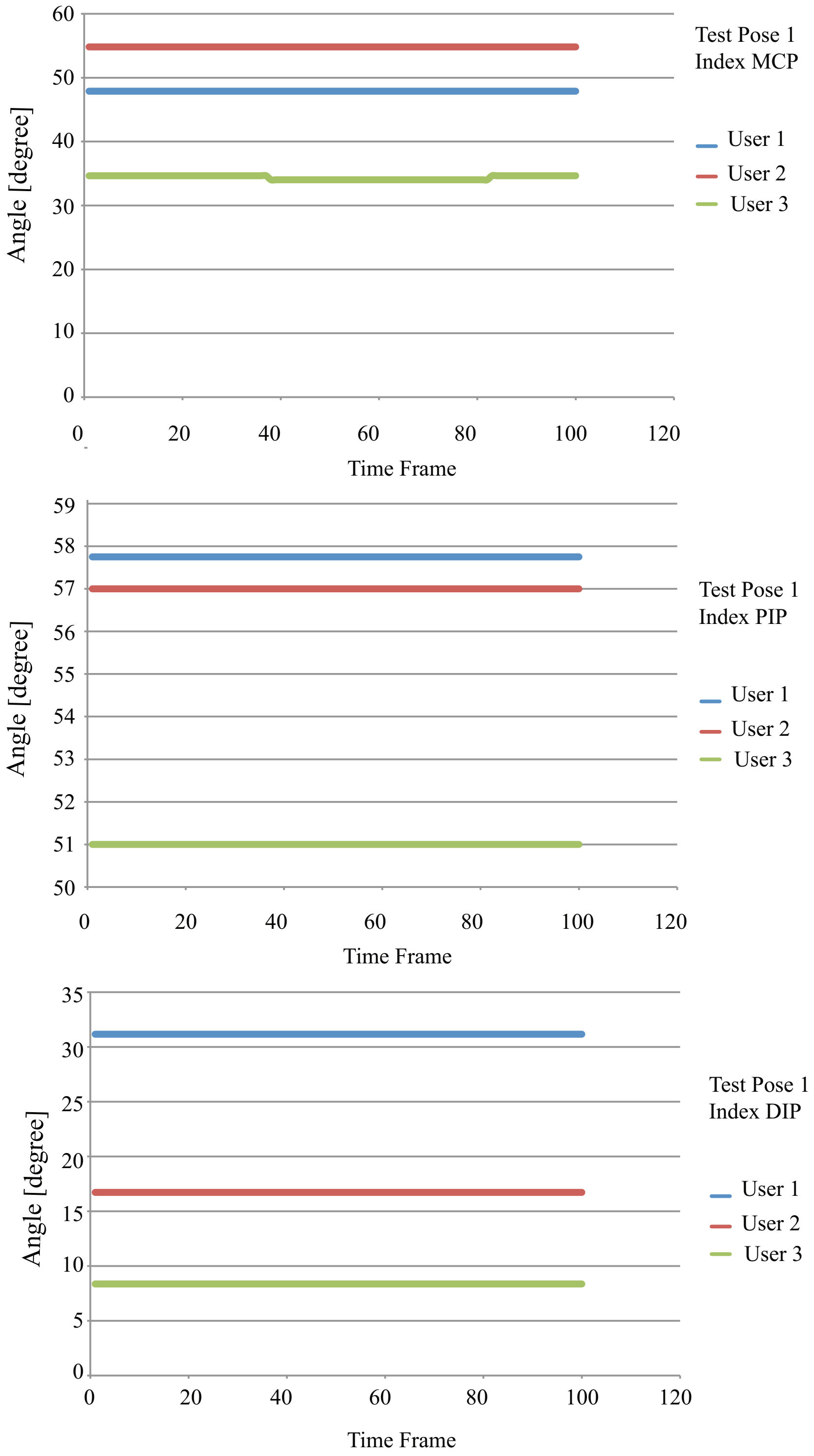


Figure 3.9: CyberGlove measurement when grasping a small paper cylinder.

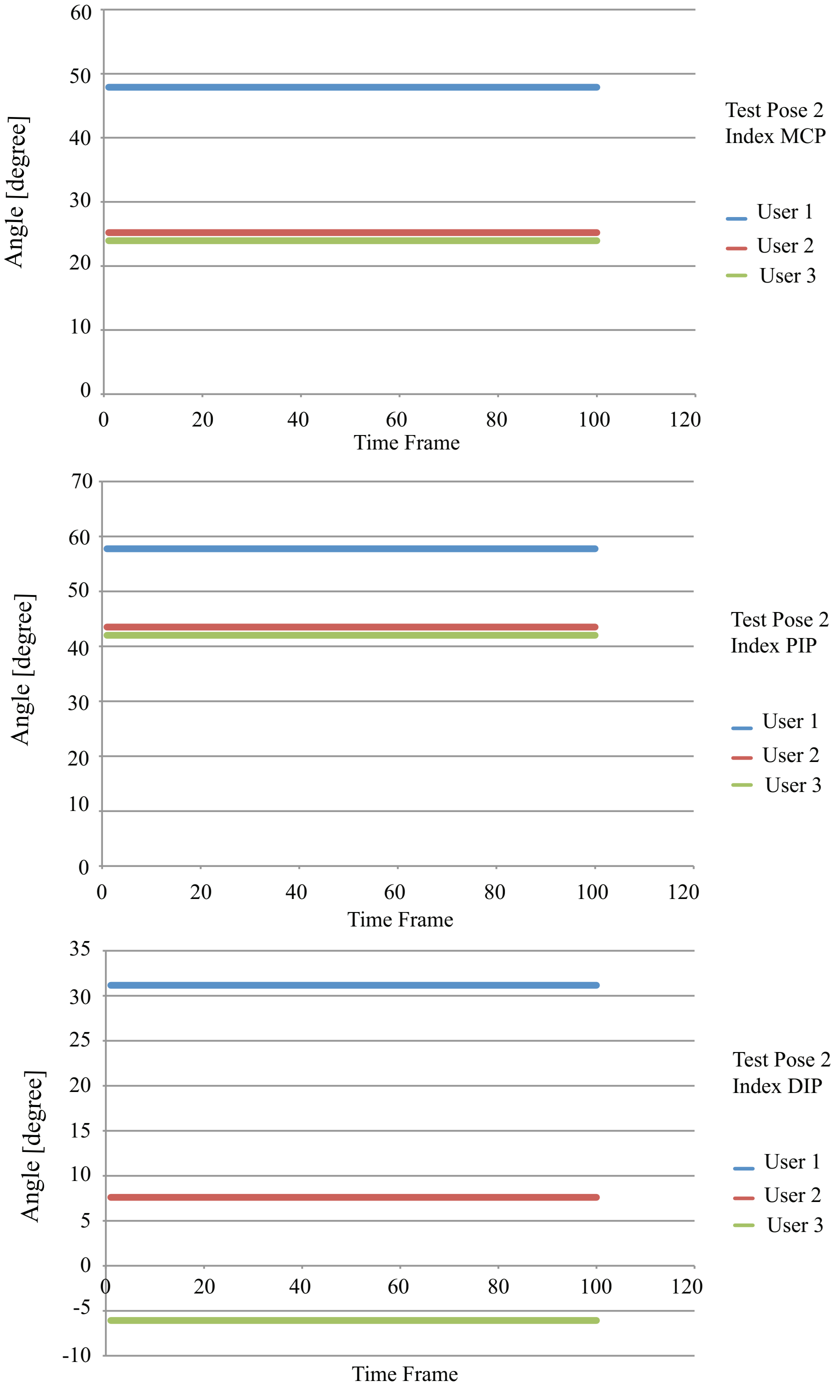


Figure 3.10: CyberGlove measurement when grasping a plastic bottle.

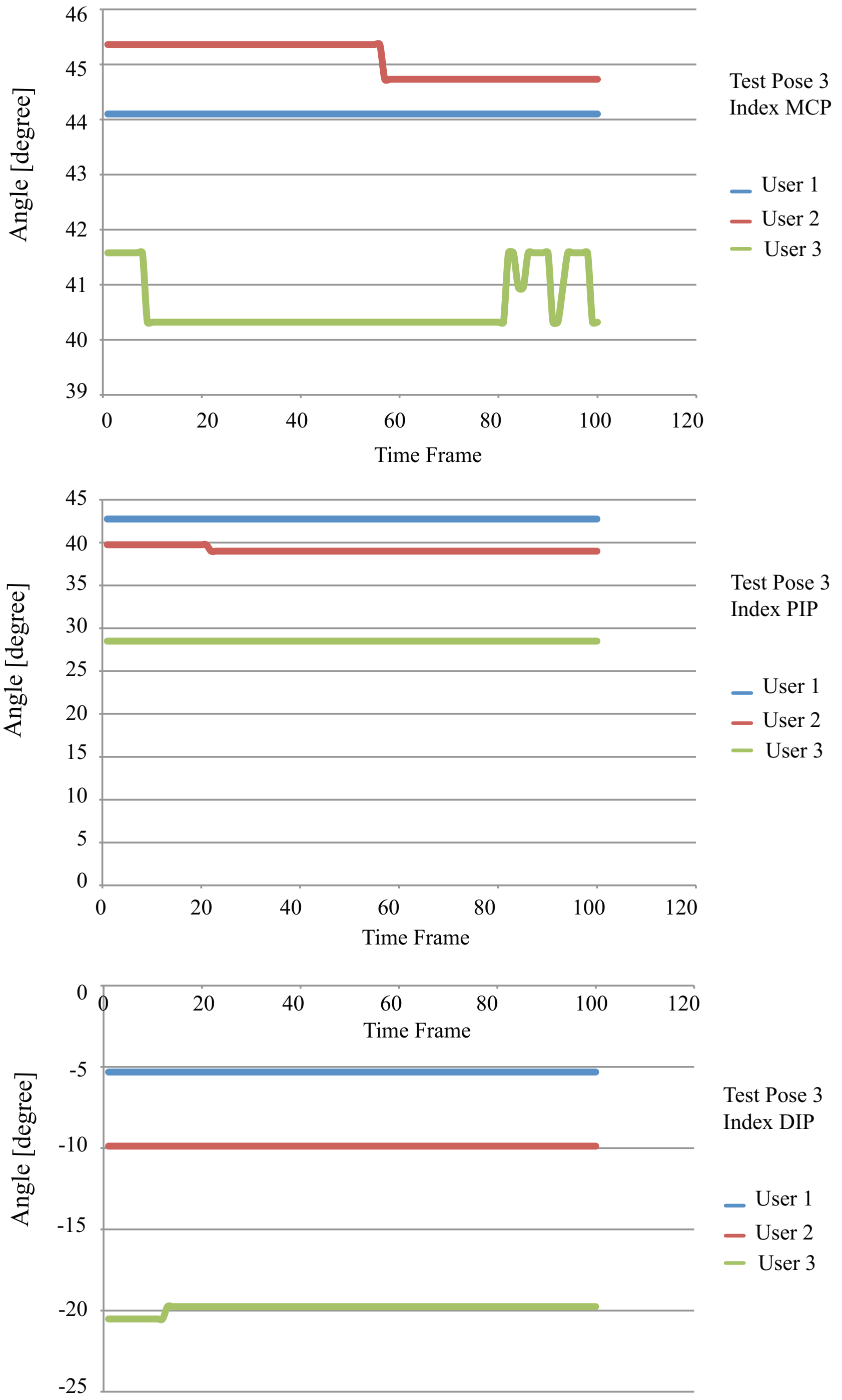


Figure 3.11: CyberGlove measurement when grasping a hardcover book.

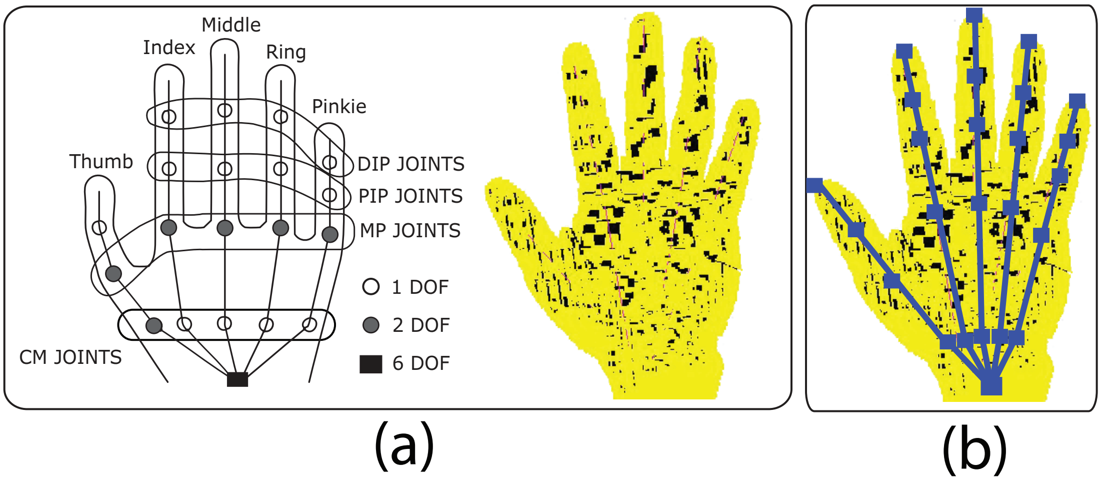


Figure 3.12: Skeletal hand model. (a) Finger bone links make up the hand skeleton and the surface structure represents skin. (b) Shape structure with underlying finger links.

shape, as shown in Figure 3.12. The new model differs from the uncalibrated model (Figure 3.4) with its surface structure. The hand model has a total of 31 degrees of freedom (DOF): six for the wrist position and palm orientation and 25 for the finger joint angles. Each finger has five degrees of freedom: two for the CM, two for the MCP, and one for the PIP. The two DOFs each of the CM and the MCP helps accommodate a wider range of hand poses. The surface structure is composed of voxel data with adjustable resolution. In our experiments, we implemented a 2×2×2 millimeter resolution per voxel unit or octant. Individual links are associated with specific surface segment and applying force to the links moves the fingers and changes the hand model shape.

3.4.2 The Hand Model Calibration

The hand model is calibrated in two steps: surface structure calibration and link structure calibration.

The voxel data of the hand at initial pose, with open palm and fully extended fingers, is obtained using shape-from-silhouette 3D reconstruction [49]. This voxel data becomes the surface structure of the hand model.

Prior to calibrating the link structure, calibration motion for each finger is

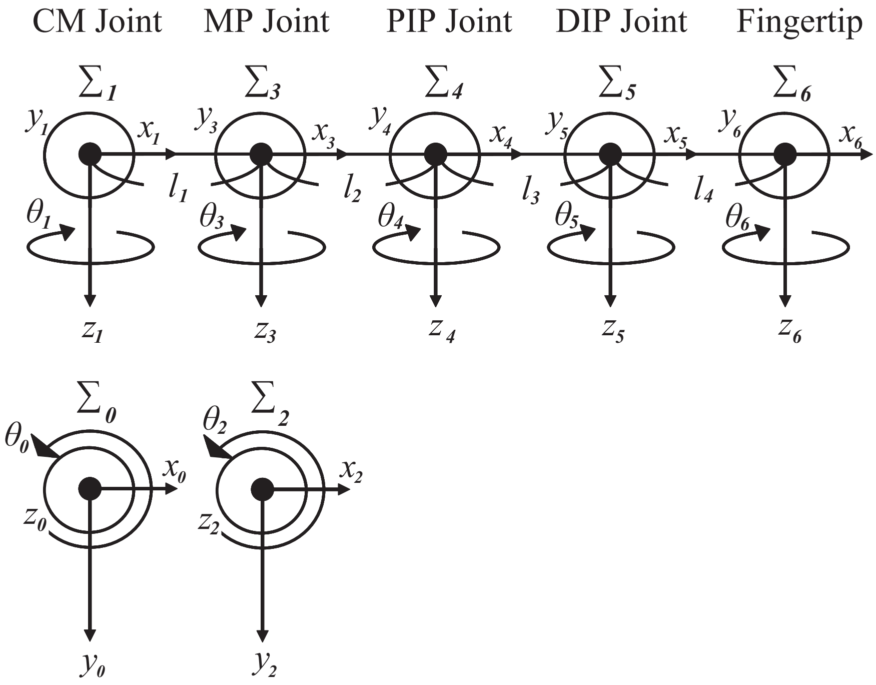


Figure 3.13: The finger linkages.

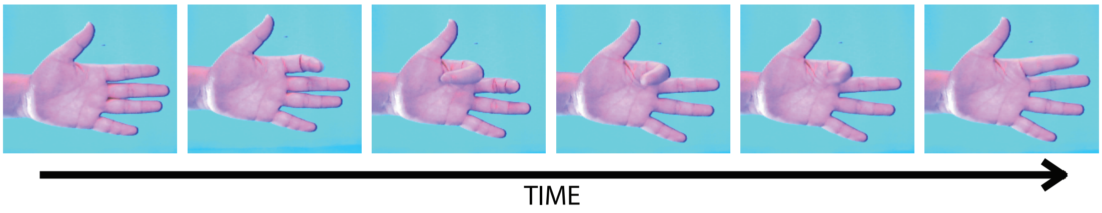


Figure 3.14: The calibration motion: each finger has its own calibration motion which consists of the finger bending toward the palm.

obtained. Then the link structure is calibrated for each finger as follows:

1. Convert the calibration motion to a time series of voxel data.
2. Manually assign initial values to the link lengths x^n_{init} where n is the number of links.
3. For the voxel data at frame k:
    (a) For every link i:
        i. Generate link length combinations within the range $\{x^i_{init} \pm 4[\text{mm}]\}$.
        ii. For every joint j:
            A. Generate joint angle combinations within the range $\{\theta^j_{a_k} \pm 5$ [deg]$\}$.
            B. Compute the cost function (Eq.3.2).
            C. Go back to Step 3(a)ii until all joints are completed.
        iii. Go back to Step 3(a)i until all links are completed.
    (b) Repeat for the next voxel data of $(k+1)$th frame from Step 3.
4. Search for the evaluated link length combinations yielding the maximum value for the cost function.

The value θ_{a_k} is further described in Section 3.4.3.

In Step 2, the joints of the model are interactively located through a graphical user interface. Then the distances between the located joints are calculated and assigned as the initial values of the link lengths.

This is repeated for other fingers using their respective calibration motions. Combining the surface structure and link structure calibration results yields the fully calibrated hand model.

A pose estimation is considered correct when the computed pose of the hand model and shape of the observation (voxel data) are very similar. This same principle applies in determining the most suitable link lengths in calibrating the hand model, since the optimum lengths should make the shape of the model, at a given pose, very similar to the observation.

The following cost function evaluates fitness between the hand model and the voxel data at each frame k:

$$f(M(x,\Theta),V_k) = \frac{M(x,\Theta) \cap V_k}{M(x,\Theta)} \tag{3.1}$$

where x is the set of link lengths $(x^1, x^2, \cdots, x^n)$ and Θ is the set of joint angles $(\theta^1, \theta^2, \cdots, \theta^m)$. n is the total number of links and m is the total number of DOF. M is the resulting hand model after its modification based on x and Θ. The cost function is the number of voxels common to model M and input data V at frame k, divided by the total number of voxels of the model. In other words, the cost function computes for the intersection of the voxel data and the hand model.

All cost function evaluations obtained during link calibration are stored and maximized:

$$x = \arg\max_x(\max_k f(M(x,\Theta),V_k)) \tag{3.2}$$

Eq. 3.2 is maximized by searching for the combination of link lengths yielding the highest value for Eq. 3.1 among all the link combinations generated for the whole calibration. The search space contains more than 64,000 link length combinations generated by the end of the calibration step. The best combination of link lengths is determined by a simple brute force method, straight-forward strategy.

3.4.3 Optimization

In maximizing Eq. 3.2, simple brute-force search among all the values generated from the cost function evaluation is employed. In order to search more efficiently, i.e., search for a manageable number of values, we must restrict the number of joint link and joint angle combinations generated per frame k.

First, five sets of calibration motion, one for each finger, are initially obtained, allowing independent calculation of the individual finger link parameters. The calibration motion consists of a finger bending toward the palm as shown in Figure 3.14. The fingers are assumed to move from the minimum to maximum allowable range as detailed in literature [14]; the joints are assumed to move simultaneously.

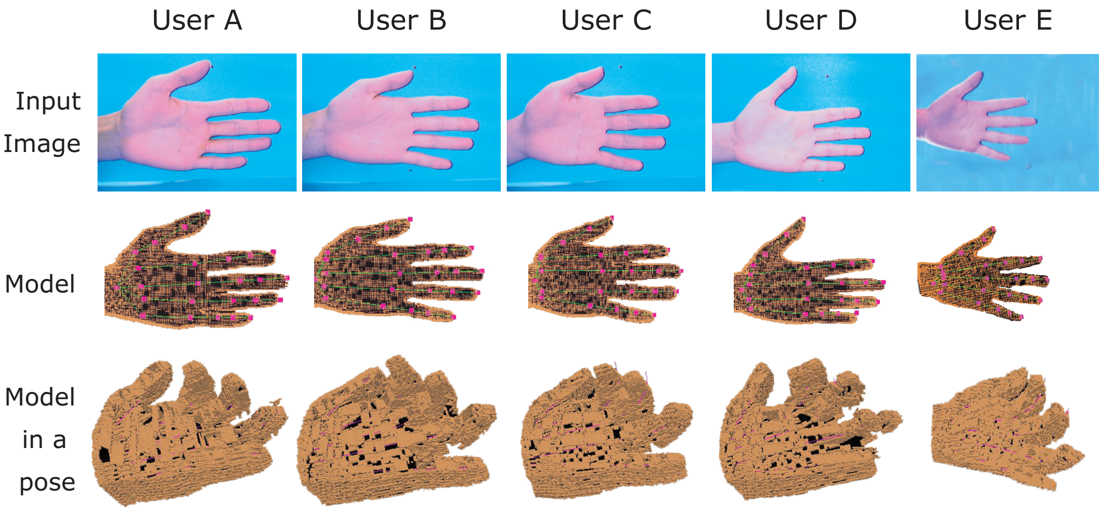


Figure 3.15: Calibrated hand model of users with different characteristics: long or short and thick or slender fingers. User D is an adult female, User E is a child, and the rest are adult males.

Assuming that each joint move at a constant velocity allowed the calculation of how much a joint would move from one time frame to the next. This gives a rough estimate of the link and joint parameters, i.e., the joint angle value θ_{a_k} at the kth frame. θ_{a_k} is computed prior to the link structure calibration step.

It is assumed that θ_{a_k} varies by around ±5 [deg] in moving from frame to frame and that the initial link length varies by ±4 [mm] for all user.

3.5. Results and Discussion

To evaluate our approach, hand models of different users were calibrated and then some of them were used in hand pose estimation experiments. The pose estimation results were compared to dataglove measurements to confirm the feasibility of the approach.

3.5.1 Hand Model Calibration

Figure 3.16 shows a hand model calibration result. Note that the shape and volume disparity between the voxel data and the hand model shown in Figure 3.1

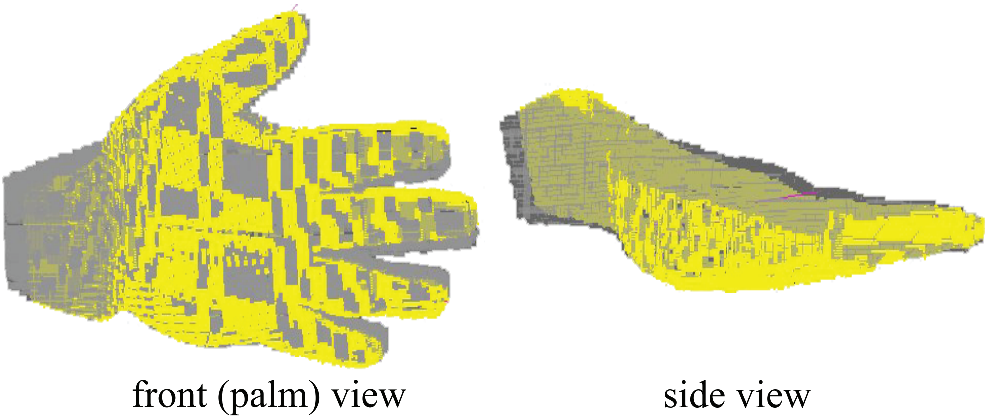


Figure 3.16: Comparison of calibrated model and observation data. The calibrated hand model (light gray pixels) and observation data (dark gray pixels) are essentially indistinguishable.

disappeared.

The proposed calibration technique was tested on the hand models of five users. The results are shown in Figure 3.15. The top row is actual user's hand and the middle row the calibrated hand model. Note the hand differences among the five users, e.g., User A's long fingers versus User C's stubby fingers, User D's slender feminine fingers versus those of the other users, who are all male, and User E's diminutive nine-year old child's hand. The bottom row shows the same pose rendered by the different calibrated models using a virtual motion simulator. Note the clear differences in the shape generated by each user hand model.

$$RelativeError = \frac{l_{manual} - l_{estimate}}{l_{manual}} \times 100 \tag{3.3}$$

To quantitatively compare calibration results, link lengths were measured manually and used as ground truth data. User hands were photocopied and the possible joint locations were identified, as illustrated in Figure 3.19. The joints were assumed to be directly beneath finger skin folds for DIP and PIP and palm bumps for MCP and CM. The dots are the locations of the link joints. The lengths between the finger joints measured were CP to MCP, MCP to PIP, PIP to DIP, and DIP to the finger tip.

Figure 3.17 compares estimated and manually measured link lengths for the

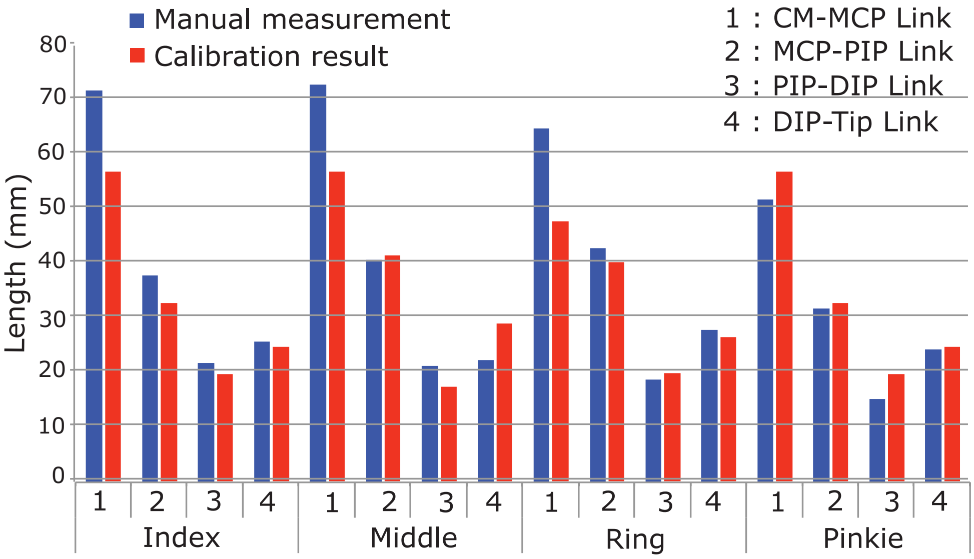


Figure 3.17: Finger link calibration results. The average error per link is 6.83 millimeters.

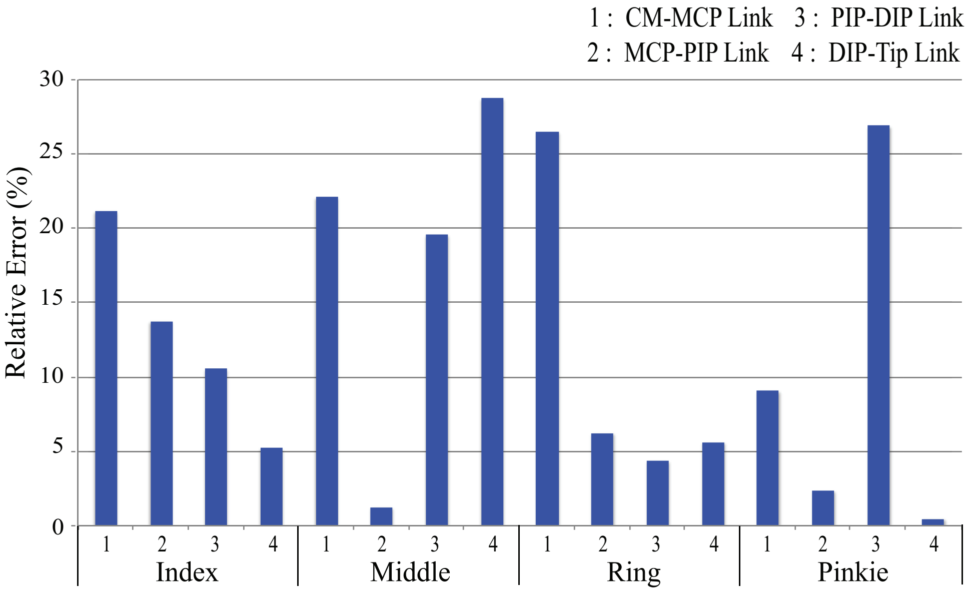


Figure 3.18: Error of the model calibration results relative to manual measurement.

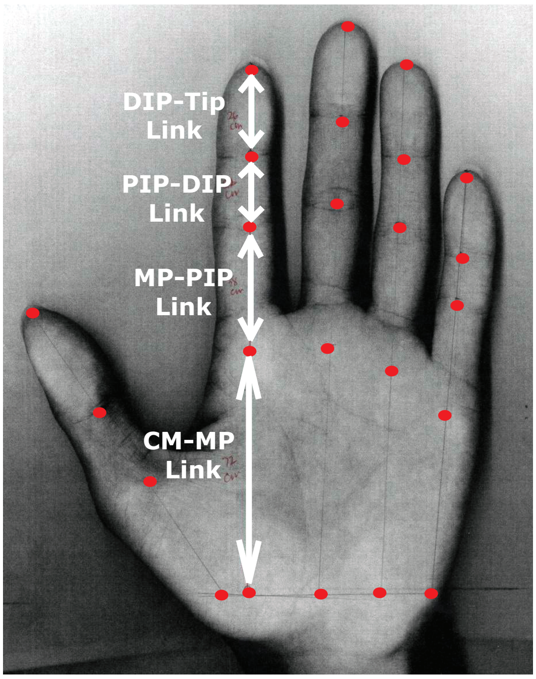


Figure 3.19: Manual measurement of the link length. The dots are the possible joint location with distances between them measured.

five users, illustrating their similarity. Estimation error is broken down in Table 3.2.

Table 3.2: Calibration Error

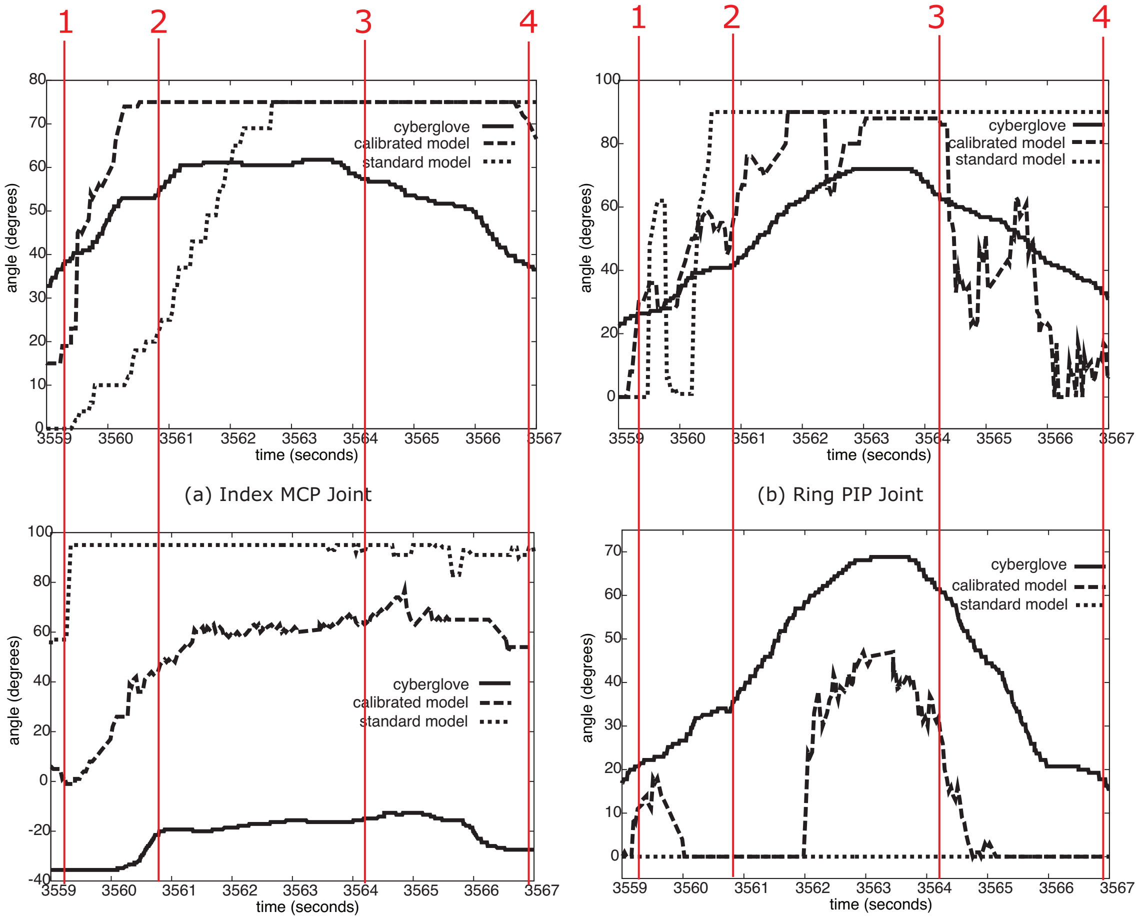
Error [mm]	**Link Percentage (%)**
< 5	68.8
5–10	12.5
>10	18.7

Compared to manual measurement, 68.8% has an error of less than 5[mm], 12.5% an error of 10[mm] or less, and 18.7% an error of more than 10[mm]. The average calibration error per link is 6.83[mm]. The large error of more than 10[mm] between the CM-MP joints, shown as label ”1” in Figure 3.17, is due to the difficulty in determining the actual location of both joints. Unlike the PIP and DIP, whose sites are clearly indicated by skin folds, CM joint locations have no clear skin indicators.

3.5.2 Hand Pose Estimation

Hand pose estimation experiments were conducted using individualized and uncalibrated hand models. The user's hand was also fitted with a dataglove (CyberGlove) to compare our approach to an input method commonly used in robotic systems. This is possible because the proposed approach only requires the hand shape. As long as hand shape is obtainable, even if it has been altered due to the gloves, model calibration and pose estimation are still possible.

Calibrating the hand model for individual users enabled different motions to be estimated. It also allowed different users, including children, to use the system. Figure 3.21 shows the closing and opening motion of a hand wearing CyberGlove. The number above each column corresponds to the numbered lines in Figure 3.20 – the time when the images were taken. Figure 3.22 shows the hand in a random finger motion. Figure 3.23 is the pose estimation for a nine-year-old's hand with two fingers closing and opening. In Fig.3.22 and 3.23, the top row is input images and the bottom row is estimation results using the calibrated hand models.



(a) Index MCP Joint

(b) Ring PIP Joint

(c) Pinkie MCP Joint

(d) Pinkie PIP Joint

Figure 3.20: Performance comparison between calibrated model, uncalibrated model, and CyberGlove measurement. Numbered lines (1-4) are points at which the images in Figure 3.21 were taken.

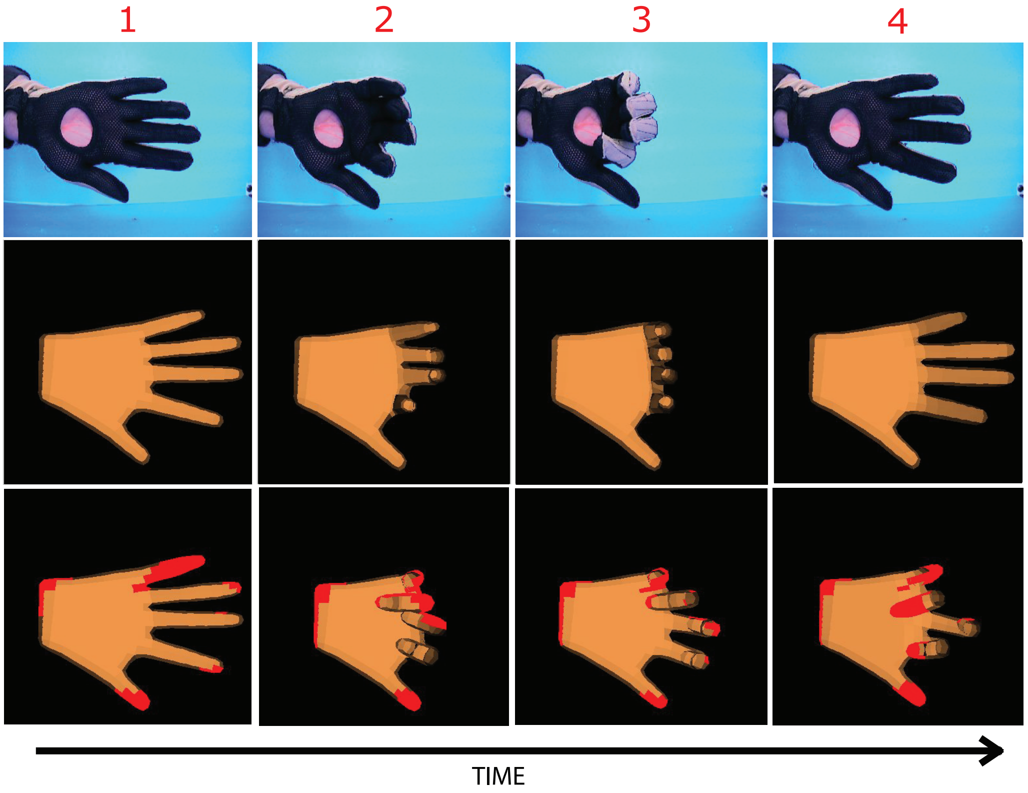


Figure 3.21: Graphical data of the performance comparison: Images taken at the numbered points in Figure 3.20. Upper row: actual CyberGlove motion images. Middle row: pose estimation result for calibrated model. Bottom row: pose estimation result for uncalibrated model. The red areas of the hand at the bottom row lie outside the observation data. For comparison, the calibrated hand model estimation result was rendered using the uncalibrated model to emphasize the difference in results.

Evaluation: Similarity in Motion Estimation

Table 3.3: Correlation Data of the Pose Estimation Comparison using CyberGlove, calibrated model, and uncalibrated model shown in Figure 3.20.

	Correlation
Index MCP	
CyberGlove - Calibrated model	-0.3729
CyberGlove - Uncalibrated model	0.4356
Ring PIP	
CyberGlove - Calibrated model	-0.0026
CyberGlove - Uncalibrated model	-0.1802
Pinkie MCP	
CyberGlove - Calibrated model	0.8246
CyberGlove - Uncalibrated model	0.2205
Pinkie PIP	
CyberGlove - Calibrated model	-0.0578
CyberGlove - Uncalibrated model	-0.4777

Figure 3.20 shows calibrated and uncalibrated model performance compared to CyberGlove measurement. Despite the big disparity between the CyberGlove measurements and the pose estimation results using either the calibrated or uncalibrated model, the pattern of the hand motion is still discernible from the graphs. Pose estimation using the calibrated model performs better than using the uncalibrated model. In Figure 3.20D, for example, using the uncalibrated model resulted in an estimation value of zero, while using the calibrated model yielded the ring finger's PIP joint motion pattern as measured by the CyberGlove.

The similarity of the motion pattern is confirmed by computing the correlation of the data sets as shown in Table 3.3. The strength of the similarity in the pattern is indicated by the correlation value. Thus, for example, the correlation value of the pinkie's MCP joint angle for the CyberGlove-Calibrated model pair indicates that motion pattern traced by the dataglove measurement is strongly followed by the calibrated model. Compare that to the values of the pinkie's MCP obtained for the CyberGlove-Uncalibrated model pair, which indicates weaker similarity in motion pattern.

The performance disparity between the models is due to the model-fitting process assuming that the correct pose estimate is reached when the model is fully within the voxel data or when the fitting evaluation starts oscillating by a constant magnitude. When the uncalibrated model is completely inside the larger voxel data, changes in the voxel data shape, specially if small, result in zero pose estimate and the model's pose need not be updated. This is seen as the flat part of the uncalibrated model estimation graph. On the other hand, using the proposed hand model makes the pose estimation process more robust, i.e., the shape similarity between the calibrated model and voxel data promotes hand motion tracking.

The disparity between the proposed approach and the dataglove measurement is clear in Figure 3.20(c) and (d) possibly due to differences in initial pose and the allowed motion range of each joint. In Figure 3.20(c), for example, the pinkie's MP joint moves within -20 to -40 [deg], compared to the calibrated model's 0 to 80 [deg] range.

Evaluation: Comparison with Model-fitting Method

The shape of the pose estimation of the different models were compared to the shape of the hand in the input images to further evaluate the system. Figure 3.24 shows the captured motion together with the estimation result using the calibrated model for the MCP joints. Estimation was done using both the calibrated and uncalibrated model. The motion is that of the hand closing and opening slowly.

Looking at the actual estimation values, a closer look at Figure 3.25 indicates that the calibrated model outperforms the uncalibrated model. This is possible because the uncalibrated model and the actual hand have different shapes. This disparity is highlighted during the middle part of the estimation process. Toward the middle Figure 3.25, the uncalibrated model's pose estimation result shows a plateau, while the calibrated model's shows a shape that resembles that of the dataglove. During this period the uncalibrated model, due to its smaller shape, is completely inside the voxel data. When this happens, the model-fitting technique would not work since the fitting only takes place if the model is outside

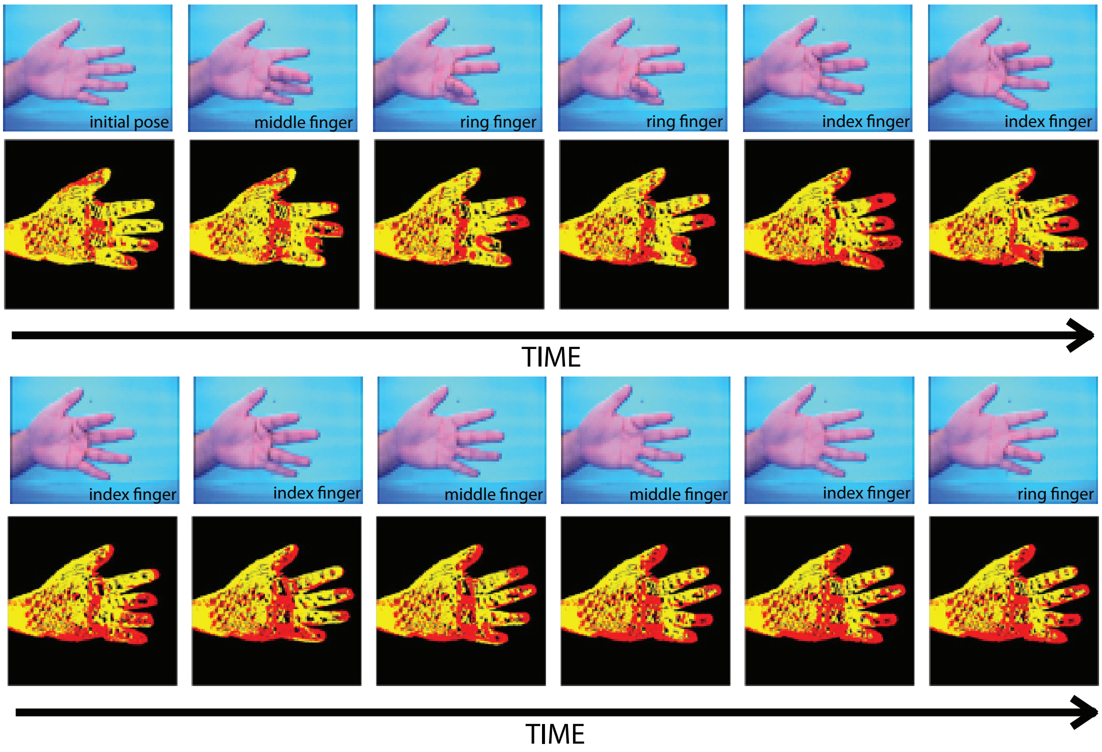


Figure 3.22: Bottom row: pose estimation result for fingers randomly closing one at a time. Top row: input images. The label in each frame indicates the moving finger.

the voxel data. Model fitting resumes when the shape of the voxel results to the uncalibrated model getting exposed again (i.e., outside of the voxel). The similarity between the calibrated model and the actual hand minimizes situations wherein the model is completely inside the voxel data.

Shape similarity comparison were done to further evaluate the improvement in using calibrated model for pose estimation. The estimation results were back-projected and their silhouette areas were compared to the silhouette area of the hand in the input image. Sampling were done at the five points indicated by the numbered red lines in Figure 3.25.

For better comparison, estimation results using the uncalibrated model were rendered with the calibrated model. Additionally, the dataglove's actual motion was also rendered with the calibrated model. The comparisons were done for each camera viewpoint illustrated in Figure 3.31. The resulting silhouettes are shown

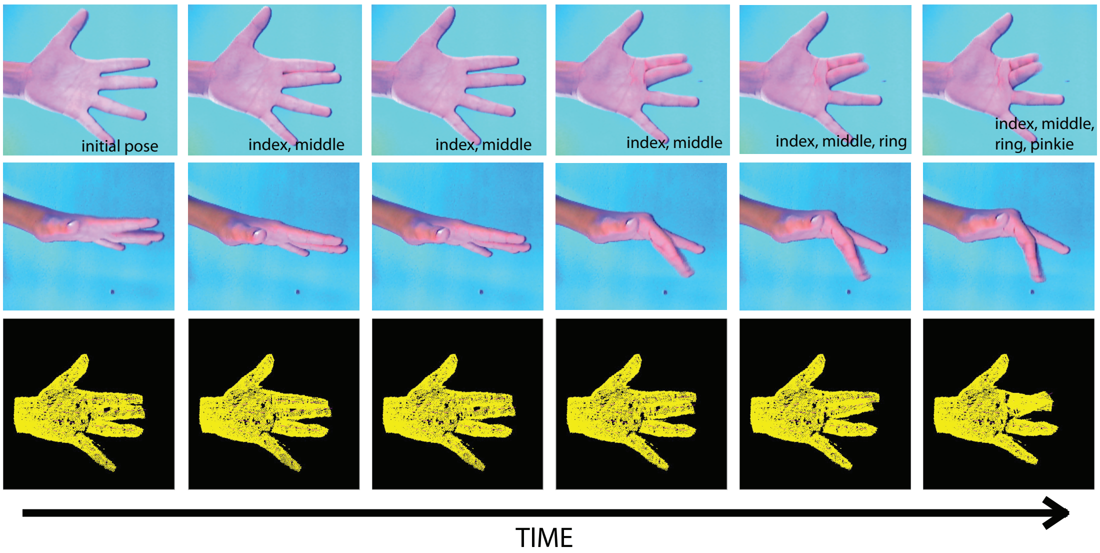


Figure 3.23: Pose estimation of a child's hand closing two fingers at a time. Top two rows: input images (front and top view). The labels in each frame indicate the moving finger.

in Figure 3.26, 3.27, 3.28, 3.29 and 3.30. For each point, the four camera views are shown for the virtual, the calibrated, and the uncalibrated model rendered with the calibrated model. The last one will be referred to as "voxel uncalibrated model", while the original uncalibrated model will be referred to as "quadric uncalibrated model".

The silhouettes of the estimation results were compared using Equation 3.4 and Equation 3.5. The AND indicates overlapping of the hand pixels at the right location and the XOR indicates disparity (i.e., non-overlapping). The best case value for the AND is 100% (complete overlapping) and the worst is 0% (no overlapping). For the XOR, the best case is 0% (complete overlapping) and the worst case is 200% (no overlapping). For easier comparison, the XOR value was normalized by dividing it by 2 such that its best case is still 0% while its worst case becomes 100%. A high AND value indicate that the silhouettes of the model and the input image are very similar in shape, while a low XOR means that the two silhouettes have minor disparity. The equations were applied between the model (estimation result) and the actual images (input) at every frame for each camera viewpoint.

$$Comparison_{AND} = \frac{SilhouetteArea_{model} \wedge SilhouetteArea_{actual}}{SilhouetteArea_{actual}} \times 100 \quad (3.4)$$

$$Comparison_{XOR} = \frac{SilhouetteArea_{model} \oplus SilhouetteArea_{actual}}{SilhouetteArea_{actual}} \times 100 \quad (3.5)$$

The result of the comparisons are displayed in Figure 3.32, 3.33, 3.34, and 3.35. Each graph compares the AND and XOR values for the virtual model, the calibrated model (CAL), the quadric uncalibrated model (Uncal-Q), and the voxel uncalibrated model (Uncal-V). The AND values are represented by the solid lines and the XOR values by the dashed lines. For all the four camera views and for most of the estimation period, the value of the AND comparison is higher for the calibrated model than for the uncalibrated model (either using voxel or using quadrics). The same goes for the XOR, where the calibrated model yielded lower values than the uncalibrated model. Unsurprisingly, the uncalibrated model using quadrics yielded the worst performance. In some cases, the value of the AND and the XOR are almost similar. Between the virtual and the calibrated model, the latter showed better shape similarity to the actual hand shape.

The average AND and XOR values for the whole range of motion are tabulated in Table 3.4. Four models are compared: the virtual, the calibrated, the quadric uncalibrated, and the voxel uncalibrated. When the voxel uncalibrated model is used for comparison, the calibrated model is more similar to the actual hand shape by 6%; when compared with the quadric uncalibrated model, the difference is 32%. With the voxel uncalibrated model, the XOR value of the calibrated model is lower by 3%, but when compared to the quadric uncalibrated model, it is lower by 14%. Thus, the calibrated model yields better estimation result than the uncalibrated model.

3.6. Conclusion and Future Work

Integrating robotic computer systems more deeply into our lives requires that needs be met for natural and non-contact interfacing systems. The hand model

Table 3.4: Comparison of Silhouettes of Estimation Results (% of the area of the hand in the input image)

	Virtual		Calibrated		Uncal - Voxel		Uncal - Quadric	
	AND	XOR	AND	XOR	AND	XOR	AND	XOR
View 1	84	17	85	14	79	16	65	23
View 2	79	15	81	13	76	16	37	35
View 3	88	20	88	14	82	20	44	33
View 4	73	25	73	21	67	23	53	28
Ave	81	19	82	16	76	19	50	30

calibration proposed uses multi-viewpoint camera and improves hand-pose estimation. A study compares the performance of CyberGlove for different users. Then hand models for five users were calibrated and compared with manual finger link measurements. The calibrated models were used to estimate the pose of a moving hand and compared the result to that of using the uncalibrated model and dataglove measurement. Using calibrated hand model produced better result than using the uncalibrated model and even enabled hand pose estimation of a child's hand, confirming the feasibility of the approach. Moreover, a comparison of silhouettes between the input image and the result of the estimation using the calibrated model shows an improvement compared to when using the uncalibrated model. When the shape of the resulting pose estimation was compared, using the calibrated model showed an improvement of up to 30% against using the uncalibrated model. Overall, the use of the calibrated model improves the hand pose estimation, making it more usable for more people and with varying motion.

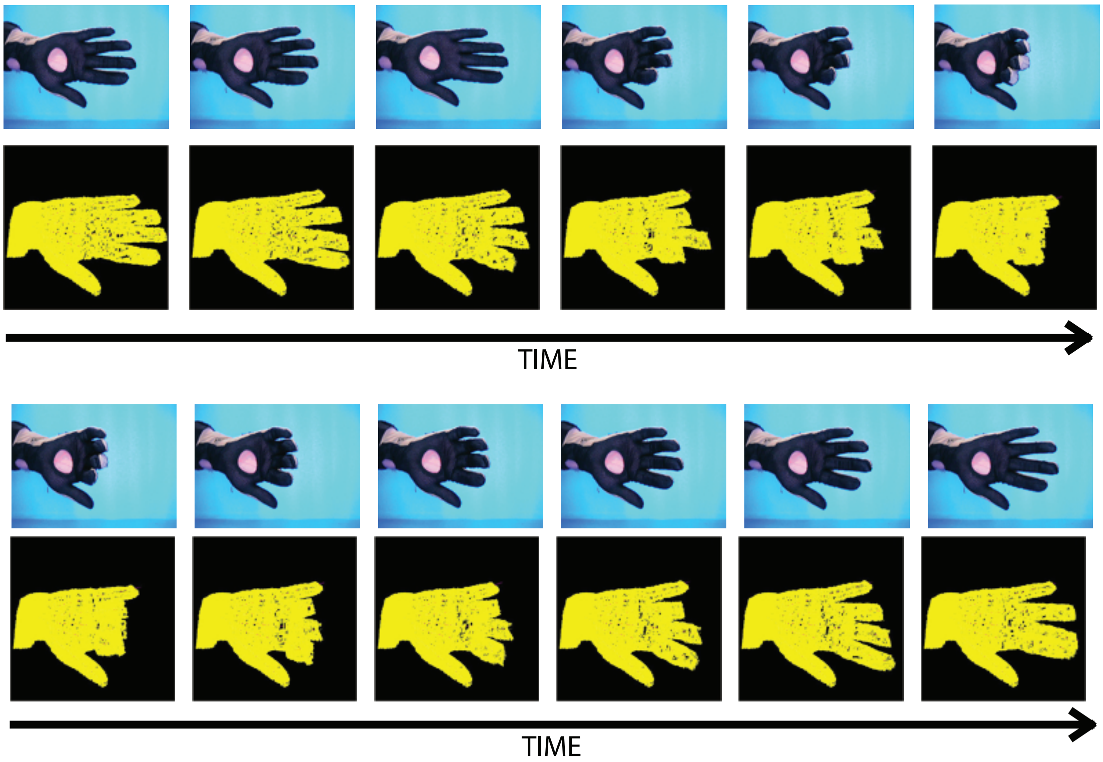
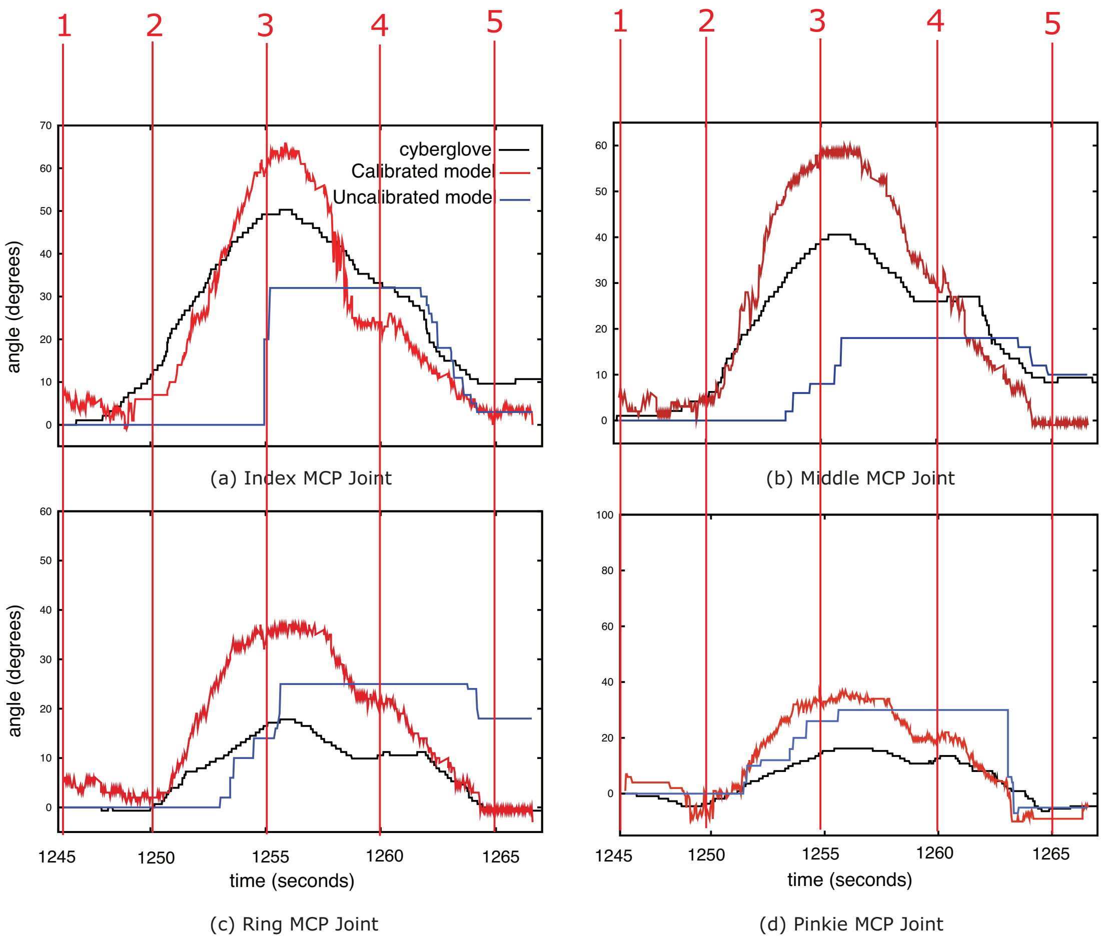


Figure 3.24: Pose estimation of a hand wearing a dataglove. Top row: input images; bottom row: estimation result using calibrated model.



(a) Index MCP Joint

(b) Middle MCP Joint

(c) Ring MCP Joint

(d) Pinkie MCP Joint

Figure 3.25: Pose estimation using the calibrated model and the uncalibrated model are compared. The numbered lines (1-5) are points at which the images in the succeeding figures were taken.

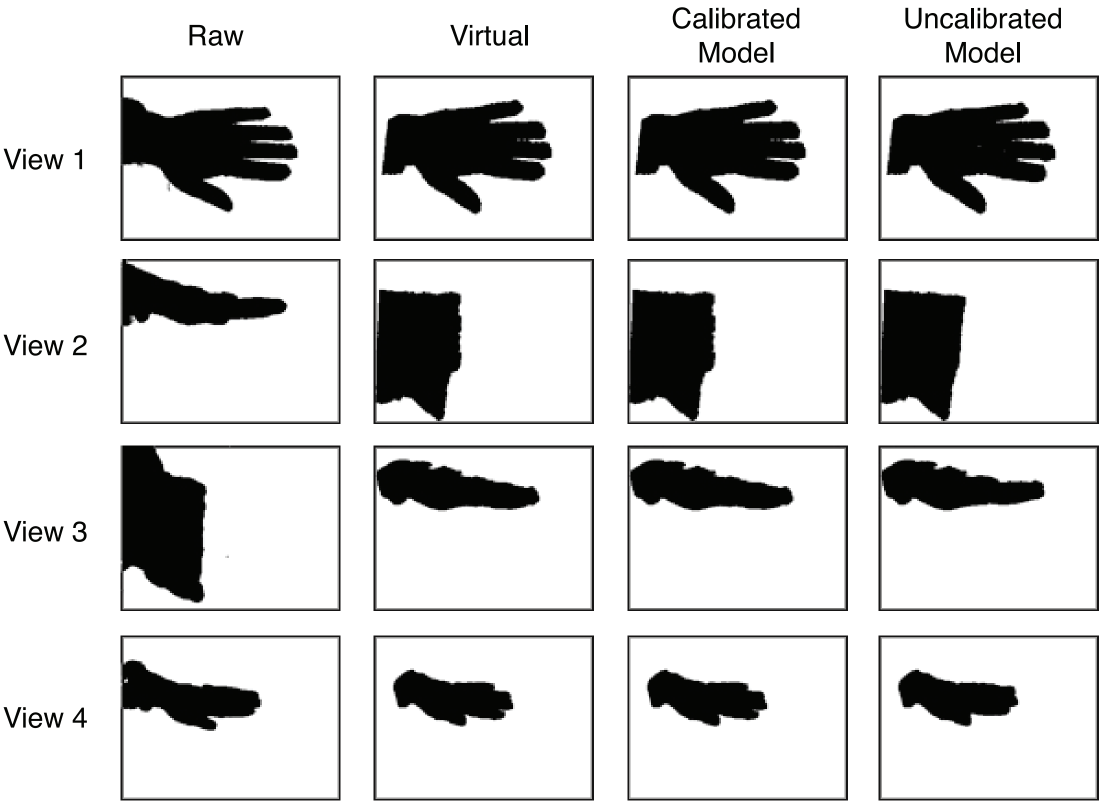


Figure 3.26: Comparison of silhouette area ratio for the virtually generated cyberglove motion, pose estimation using calibrated model, and using uncalibrated model for Point 1 of Figure 3.25. The uncalibrated model shape is rendered using the calibrated model to minimize bias in comparison.

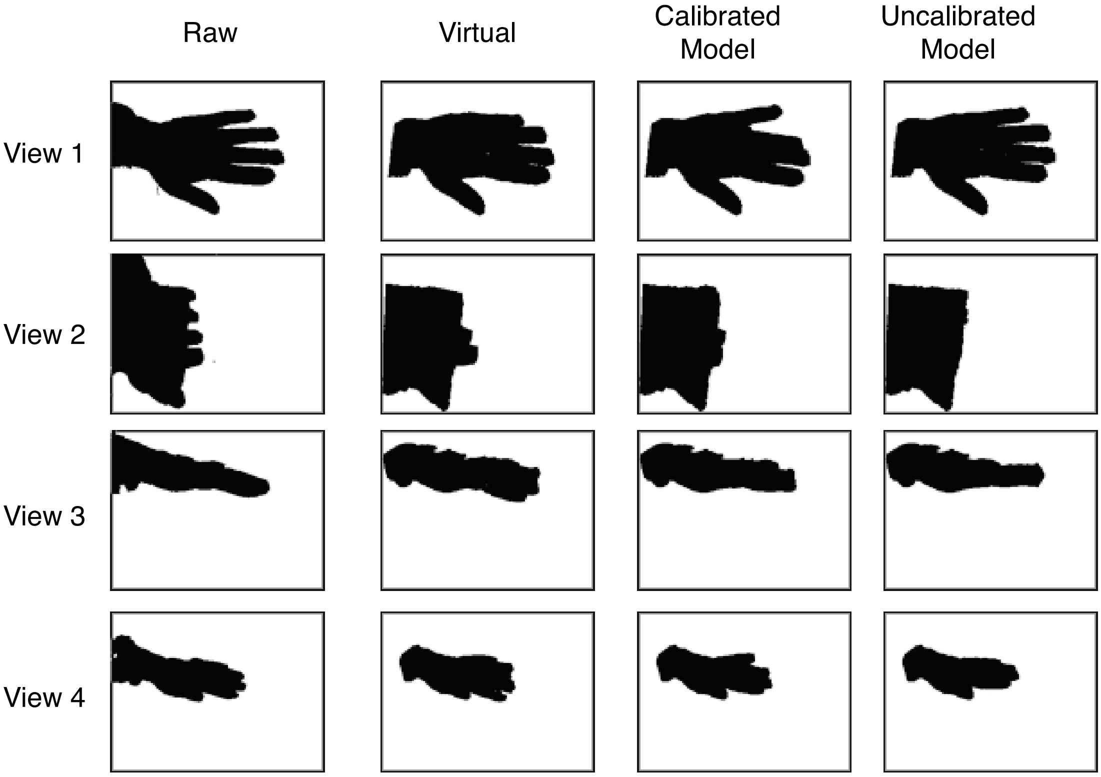


Figure 3.27: Comparison of silhouette area ratio for the virtually generated cyberglove motion, pose estimation using calibrated model, and using uncalibrated model for Point 2 of Figure 3.25. The uncalibrated model shape is rendered using the calibrated model to minimize bias in comparison.

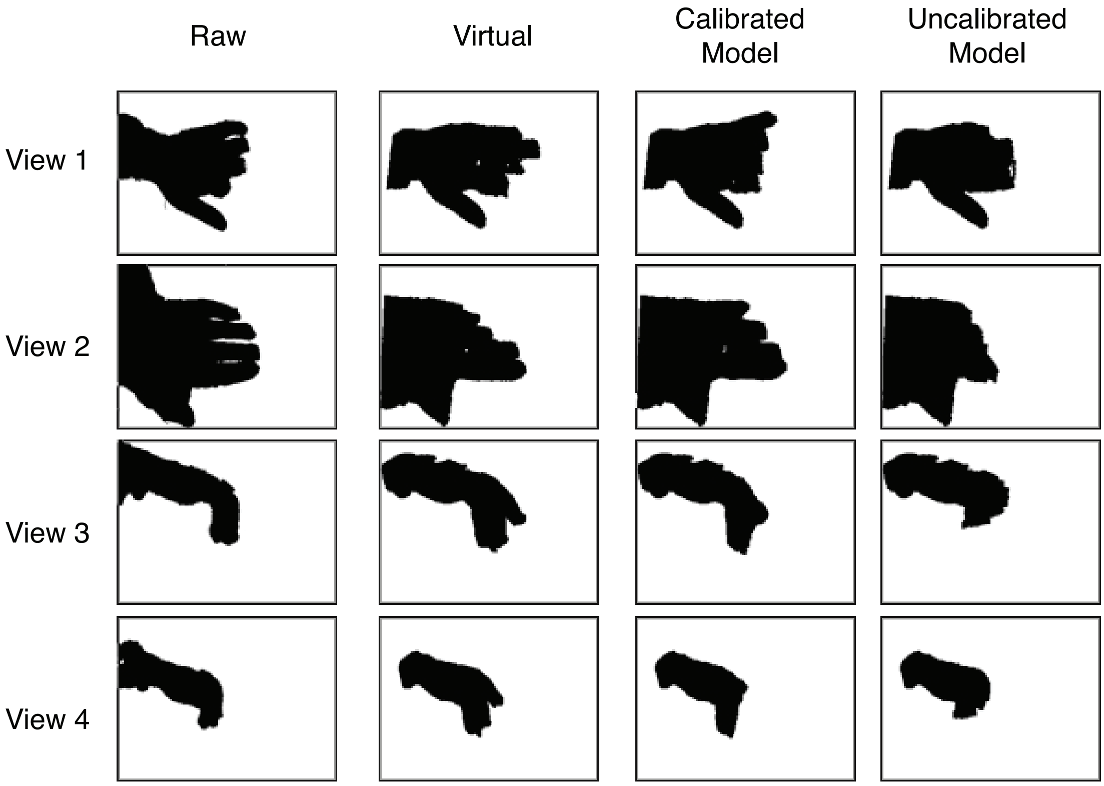


Figure 3.28: Comparison of silhouette area ratio for the virtually generated cyberglove motion, pose estimation using calibrated model, and using uncalibrated model for Point 3 of Figure 3.25. The uncalibrated model shape is rendered using the calibrated model to minimize bias in comparison.

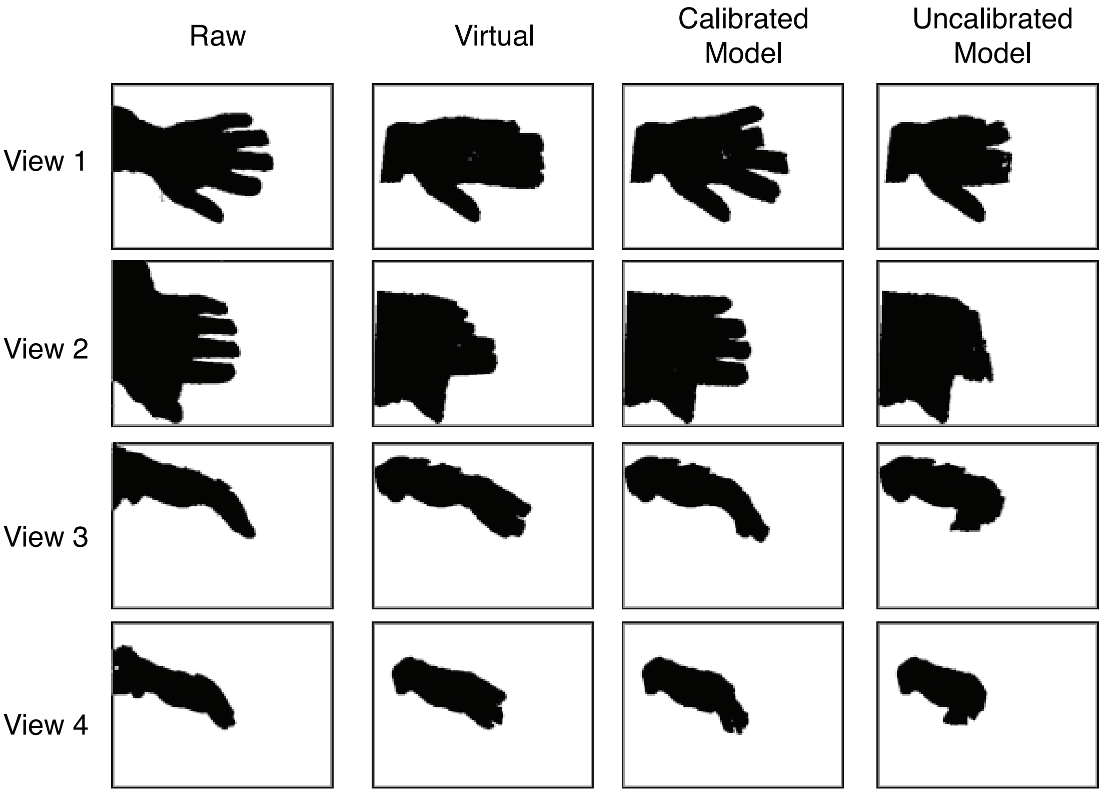


Figure 3.29: Comparison of silhouette area ratio for the virtually generated cyberglove motion, pose estimation using calibrated model, and using uncalibrated model for Point 4 of Figure 3.25. The uncalibrated model shape is rendered using the calibrated model to minimize bias in comparison.

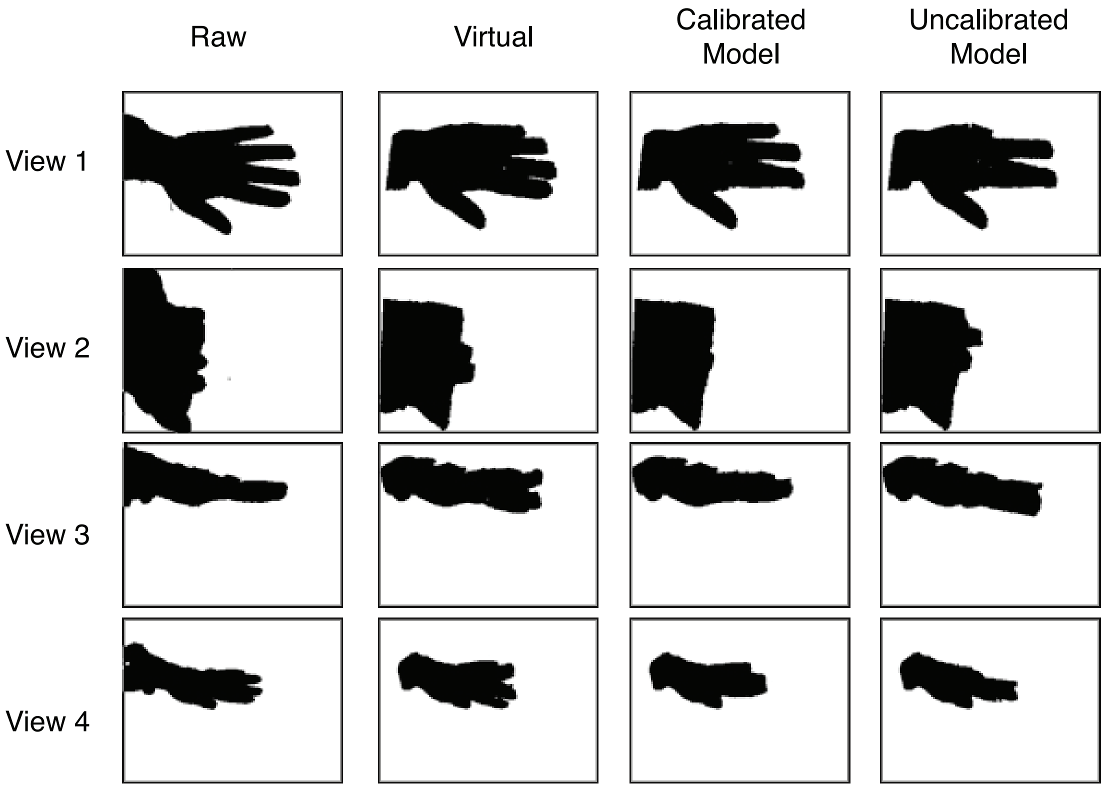


Figure 3.30: Comparison of silhouette area ratio for the virtually generated cyberglove motion, pose estimation using calibrated model, and using uncalibrated model for Point 5 of Figure 3.25.

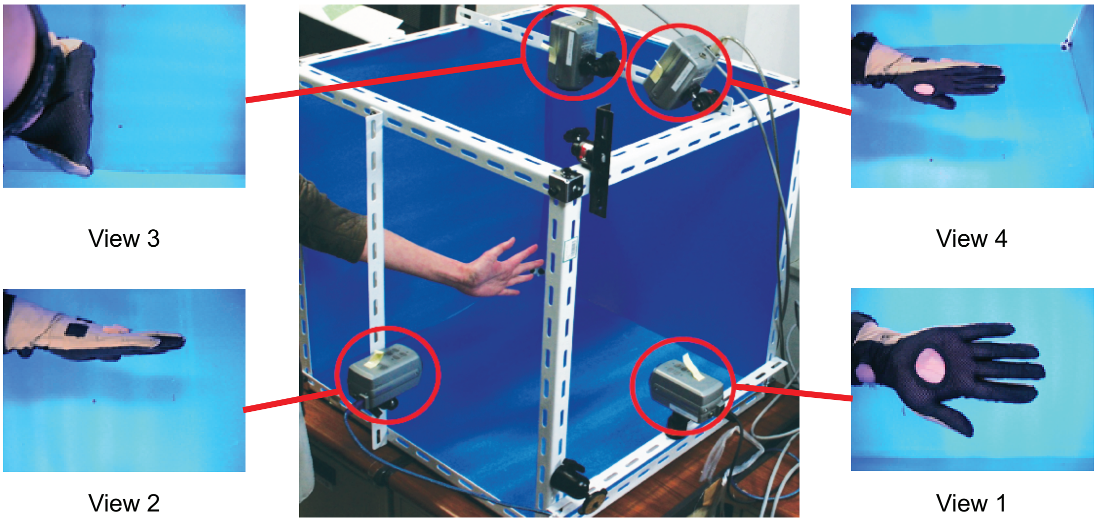


Figure 3.31: Camera view configuration showing what the cameras can see.

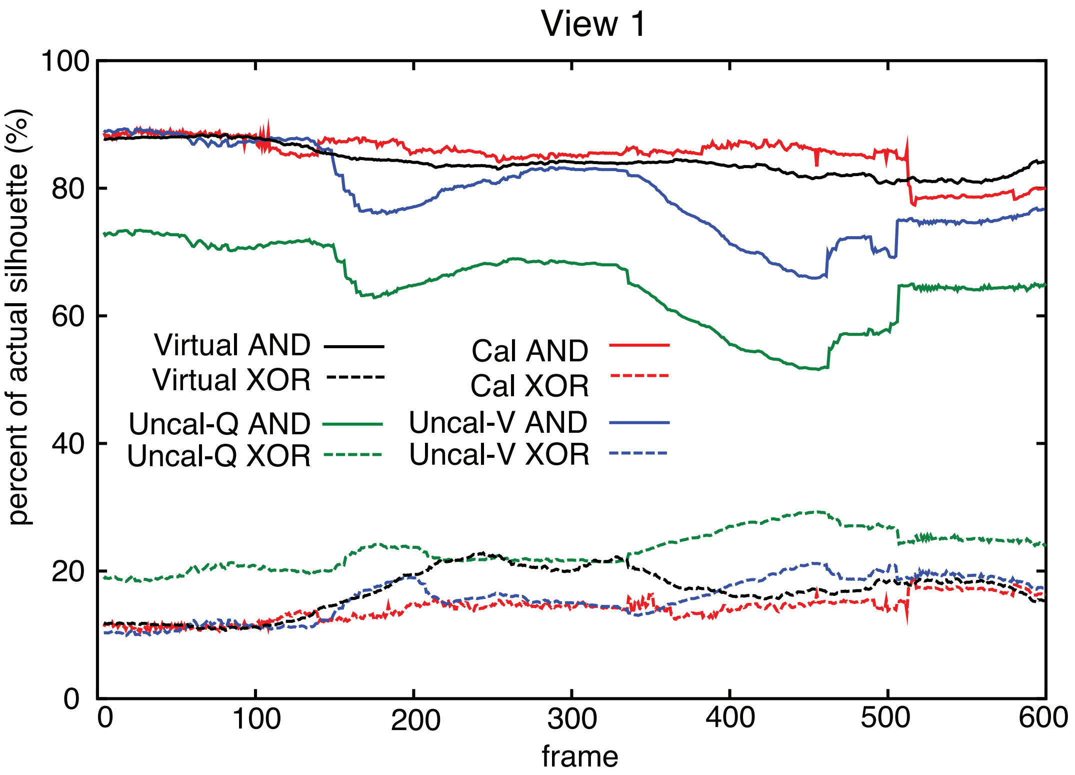


Figure 3.32: Comparison of AND and XOR of between the virtual model (Virtual), the calibrated model (Cal), the quadric uncalibrated (Uncal-Q) and the voxel uncalibrated models (Uncal-V) in Figure 3.25 for View 1. The solid lines are for the AND values and the dashed lines are for the XOR values. The calibrated model consistently yields high AND and low XOR compared to the other models.

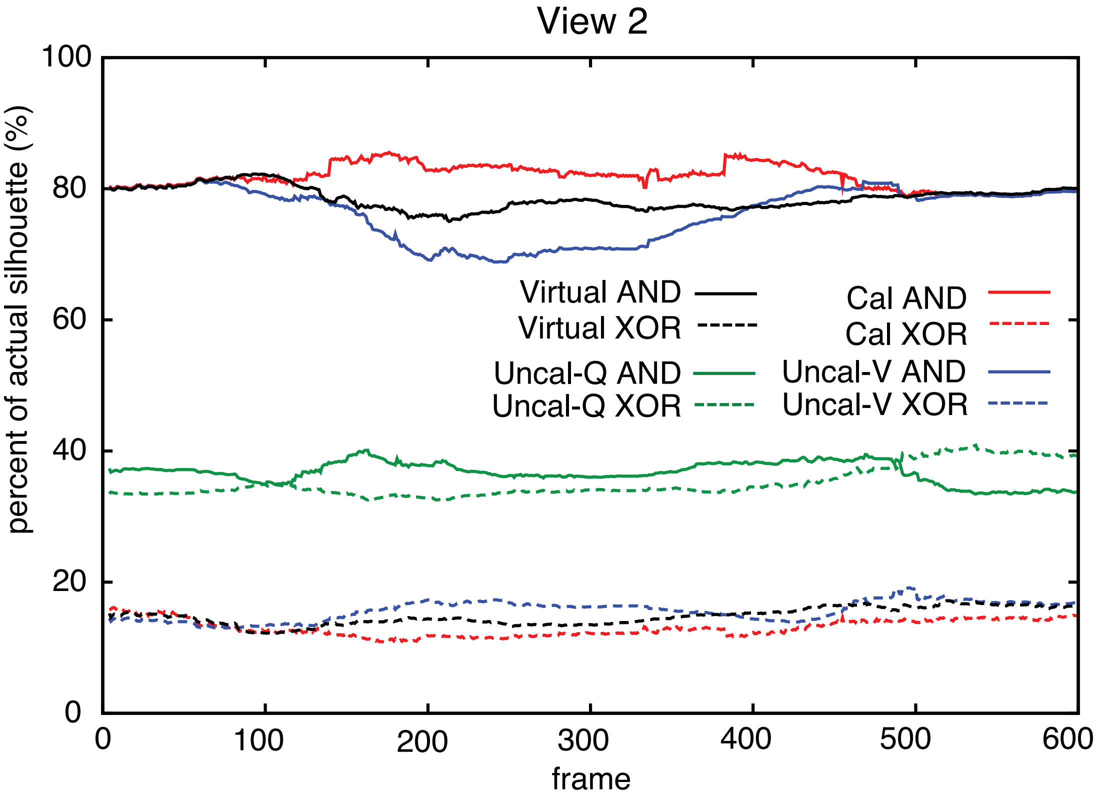


Figure 3.33: Comparison of AND and XOR of between the virtual model (Virtual), the calibrated model (Cal), the quadric uncalibrated (Uncal-Q) and the voxel uncalibrated models (Uncal-V) in Figure 3.25 for View 2. The solid lines are for the AND values and the dashed lines are for the XOR values. The calibrated model consistently yields high AND and low XOR compared to the other models.

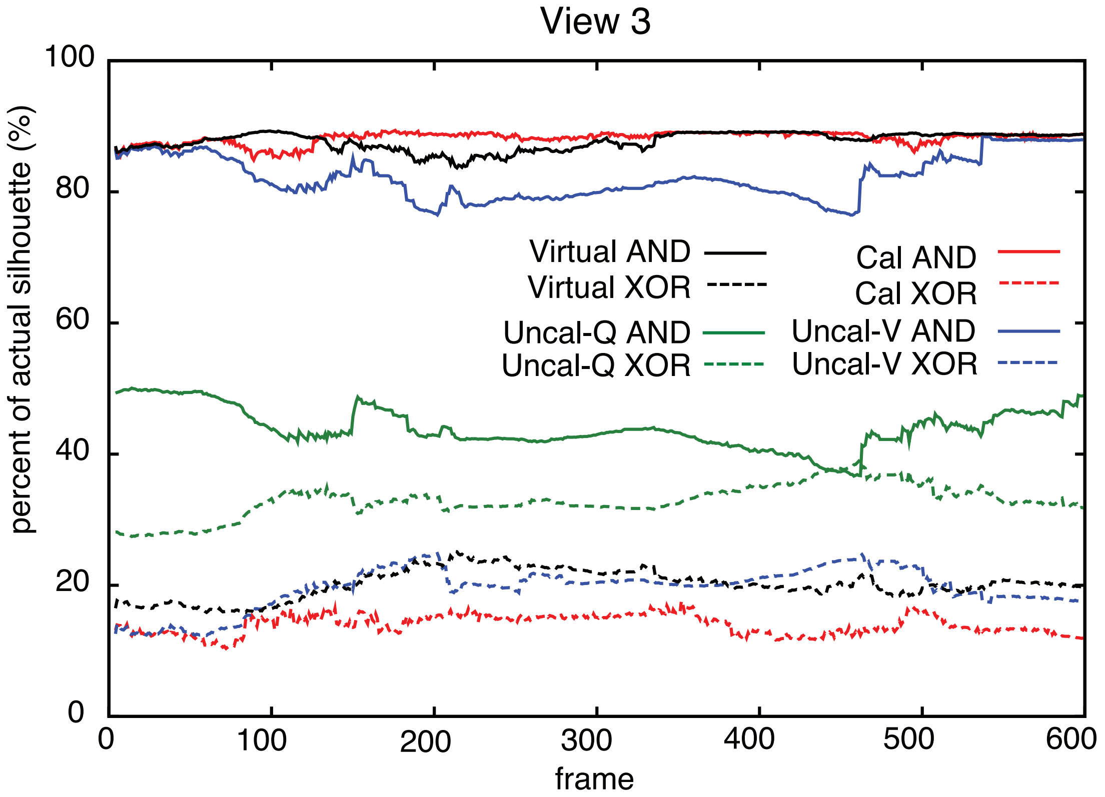


Figure 3.34: Comparison of AND and XOR of between the virtual model (Virtual), the calibrated model (Cal), the quadric uncalibrated (Uncal-Q) and the voxel uncalibrated models (Uncal-V) in Figure 3.25 for View 3. The solid lines are for the AND values and the dashed lines are for the XOR values. The calibrated model consistently yields high AND and low XOR compared to the other models.

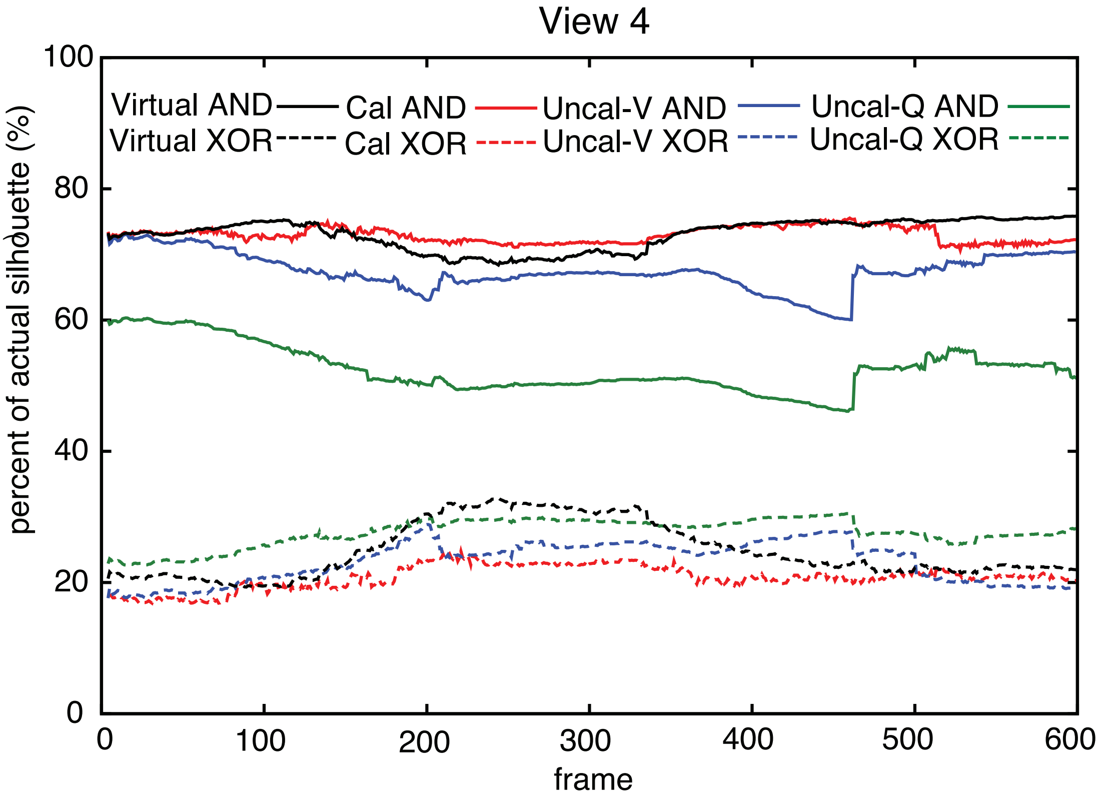


Figure 3.35: Comparison of AND and XOR of between the virtual model (Virtual), the calibrated model (Cal), the quadric uncalibrated (Uncal-Q) and the voxel uncalibrated models (Uncal-V) in Figure 3.25 for View 4. The solid lines are for the AND values and the dashed lines are for the XOR values. The calibrated model consistently yields high AND and low XOR compared to the other models.

Chapter 4

Hand Pose Estimation using Predictive Filter

> The only relevant test of the validity of a hypothesis is comparison of prediction with experience.
>
> Milton Friedman

4.1. Introduction

Motivation As discussed in the previous chapters, the over-all goal of this thesis is to create a hand pose estimation system that allows the use of natural and unrestricted hand motion as a tool for interaction. The hand pose estimation system presented in Chapter 3 is able to estimate the local pose of the hand (finger joint angles). It can also estimate the global pose of the hand (palm orientation and wrist location), but not both parameters at the same time. To truly allow a natural motion as input, the system must be able to estimate both poses simultaneously. This chapter presents a vision-based model-based approach that uses multiple cameras and predictive filtering to simultaneously estimate global and local poses of the hand.

A predictive filter is a mathematical method that recursively estimates the current internal state of a system using the estimated state from the previous time step and current noisy measurements. A good example of this is the Kalman Filter (KF), a simple form of a dynamic Bayesian network. Using a predictive filter, state parameters can be tracked over time by using noisy measurement data and a mathematical process model. In computer vision, especially in hand pose estimation, the hand parameters can be tracked or predicted using noisy measurements (input images from camera), a process model that describes how the parameters change over time (the hand model dynamics) and the relation between measurements and the hand pose (observation function).

Problem Statement In estimating the complete hand pose, i.e., both global and local parameters, the high degrees of freedom (DOF) of the hand becomes an essential issue to tackle. In the experiments presented in Chapter 3, up to 16 DOFs of local pose were able to be estimated. If the global pose was estimated at the same time as the local pose, the total number of DOFs that could be successfully tracked would be 19 or 24. Moreover, with such large numbers of DOF, the hand motion can also be formulated non-linearly, as previously noted by Stenger *et al.* [20].

4.1.1 Related Works on Pose Estimation Using Bayesian-based Filters

Hand motion tracking is not a linear problem, and predictive tracking solutions for non-linear systems are available including Extended Kalman Filter (EKF), UKF, Gaussian sum filter, particle filter (PF), and grid-based methods. Extended Kalman Filter is a straight-forward adaptation of the Kalman Filter to non-linear systems. Shimada *et al.* used EKF to estimate the pose of the hand and refine the 3D shape model even when using only a monocular camera and without any depth information [43]. A modified EKF through constraint fusion was used by Azoz *et al.* to localize and track an articulated arm [51]. Another extension of the Kalman Filter is the UKF [52] which Stenger *et al.* used to track the hand motion with using truncated quadrics for the hand model [20].

Gumpp *et al.* used particle filter (PF) to track the hand motion of the user in order to control a 20-DOF robot hand [53]. The high number of particles needed in a PF implementation made Lin *et al.* parametrized the hand configuration space to be able to use a lower number of particles and consequently speed up the computation [54]. Thayananthan *et al.* and Stenger *et al.* both used grid-based filtering to search for the representative pose by traversing the tree nodes with high probabilities [41, 24]. They were able to do a fast search because the tree nodes' probabilities were updated during tracking and the children of the nodes with small probabilities were skipped.

4.1.2 Proposed Method

Unscented Kalman Filter, a predictive filter that belongs to the Bayesian-based filter family is proposed to estimate the many DOFs of the hand and the non-linearity of the hand motion and the measurement process. The UKF estimates the pose by minimizing the error between the hand model and the observation data of the hand motion. During the minimization step, the hand model, a skeletal model with skin, is fitted to the observation data made of voxels. This chapter builds on the work described in the previous chapter, and thus, the same camera system is used.

Stenger *et al.* [20] has used UKF in tracking the motion of the hand, but this thesis differs from his work on the following: the hand model, the observation function and observation vector, and the number of DOFs estimated. Instead of truncated quadrics, a 31 DOF skeletal hand model with either quadrics (uncalibrated model) or voxel (calibrated model) for skin (surface structure) was used . To build the observation vector or the measurement, Stenger *et al.* projected the model's quadrics and used distance measurements between the silhouette edges and the outline of the projected quadrics. In contrast, 3D distance measurement was used in the proposed approach. Finally, only 3 global and 1 local parameters were estimated in [20], while the proposed method managed 3 global and 16 local parameters.

4.2. Unscented Kalman Filter (UKF)

Unscented Kalman Filter belongs to the Kalman Filter family. It is a recursive estimator that uses information from the previous time frame in addition to the current observation measurement to make an estimate of the current state. KF

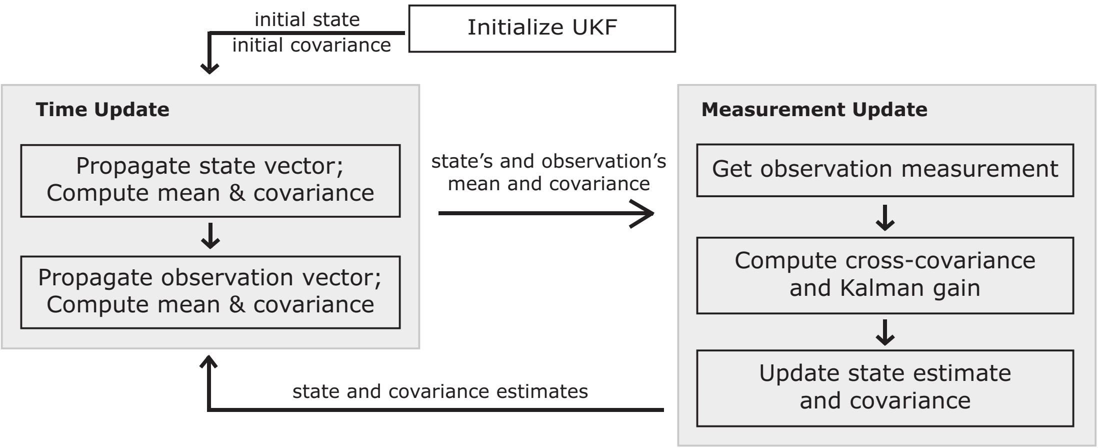

initial state
initial covariance
Initialize UKF
Time Update
Propagate state vector; Compute mean & covariance
Propagate observation vector; Compute mean & covariance
state's and observation's mean and covariance
Measurement Update
Get observation measurement
Compute cross-covariance and Kalman gain
Update state estimate and covariance
state and covariance estimates

Figure 4.1: The Unscented Kalman Filter (UKF) process. This is very similar to a Kalman Filter, except that with the UKF the initial state estimate (under the Time Update box) is obtained from the sigma (particle) propagation.

is well suited for linear systems, but for non-linear ones, EKF and UKF are available solutions. EKF linearizes about the current mean and covariance by differentiating the state and observation functions, requiring the use of partial derivatives or Jacobian.

In contrast, UKF uses unscented transformation method, which calculates the statistics of a random variable that undergoes non-linear transformation [52]. The probability density is propagated using a set of sigma particles computed deterministically. One advantage of UKF over EKF is its accuracy of up to the second order for any non-linearity. UKF is also easier to use than EKF because Jacobian derivation is a non-trivial step.

Particle filter (PF) is more sophisticated than UKF and has better accuracy than either EKF or UKF. Instead of modeling the posterior distribution as the UKF does, PF models the full posterior distribution. Its accuracy comes at a price, there should be sufficiently large number of particles in order to model the posterior distribution completely. Here, the UKF edges out PF - the former requires fewer samples than the latter.

Thus, the choice of UKF over EKF and PF is a trade-off between speed and ease of implementation and accuracy. Xiong *et al.* studied the performance of UKF under certain conditions and showed that it performs robustly in general tracking applications of non-linear systems [55].

Figure 4.1 shows the overview of the UKF process, which is composed of two main parts, similar to a KF. First is the time-update, wherein the initial state estimate is computed by selecting sigma points and solving for its mean and covariance. The observation is also propagated in this step and its mean and covariance are also calculated. The second part is the measurement-update. The Kalman gain and cross-covariance of the propagated state and the propagated observation are calculated and used to update the state and its covariance. The computational details are discussed in Appendix D.

4.3. Hand Pose Estimation using Multi-viewpoint Cameras

The same hand pose estimation system described in Section 3.2 is used in this chapter. Both uncalibrated and calibrated hand models are used. The uncalibrated model has quadrics for its surface structure, while the calibrated model has voxels for surface structure. For both models, the models have 19 joints, 31 DOFs, and 24 links. The observation data is also similar: images from the cameras converted to voxel by shape-from-silhouette technique [49].

Ueda *et al.* minimized the error between the voxel data and the skeletal model using virtual force [22]. Although their technique has the advantages of being simple and fast, it cannot estimate the global and local poses simultaneously. It also has difficulty in recovering from erroneous estimation. In the proposed approach, the global and the local parameters are estimated simultaneously using UKF. Estimating the finger motion, palm rotation and wrist translation improves the flexibility of the hand pose estimation system: it can accept a more dynamic hand motion as input.

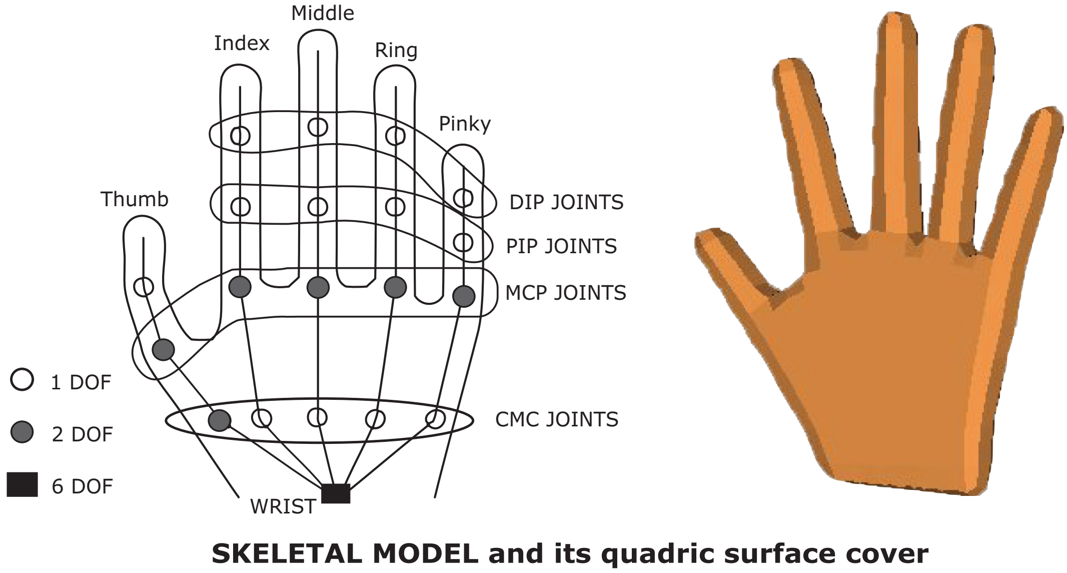


Figure 4.2: The uncalibrated hand model. The skeletal model is covered with quadric surfaces.

4.4. UKF in Hand Pose Estimation

4.4.1 Overview of the Unscented Kalman Filter

This section presents how UKF was used to estimate the hand pose. UKF was chosen over EKF because of its simpler implementation and over particle filter because of fewer particles needed. It is also accurate up to the second order [52]. Lastly, the relationship between the observation data (camera images converted to voxel) and the hand pose is non-linear.

For the hand motion, the state dynamics describes the change in the hand shape from one time frame to the next:

$$\mathbf{X_k} = f(\mathbf{X_{k-1}}, \mathbf{R_k}) \tag{4.1}$$

where:
f is the system dynamic,
$\mathbf{X_k}$ is the state vector of size n at time k, and
$\mathbf{R_k}$ is the state noise covariance.

The state variables, the hand pose parameters, are propagated deterministi-

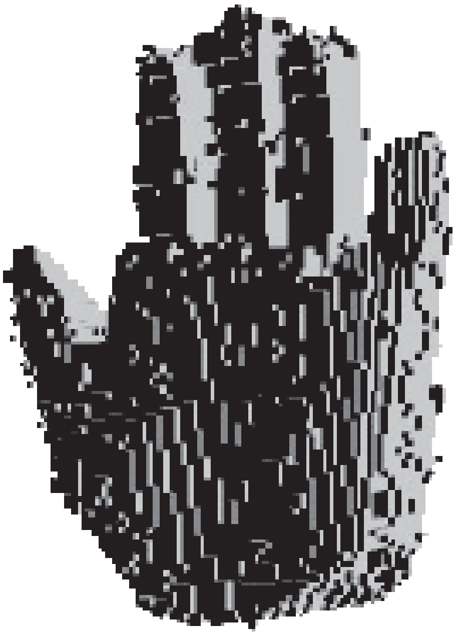


Figure 4.3: The voxel data is derived from the silhouettes of the input images.

cally using Equation 4.1 and the following:

$$\mathbf{X_k^i} = \begin{cases} \mathbf{X_k^0} = \mathbf{\bar{X}_{k-1}} \\ \mathbf{X_k^i} = \mathbf{\bar{X}_{k-1}} - (\phi)_i & i = 1, \ldots, n \\ \mathbf{X_k^i} = \mathbf{\bar{X}_{k-1}} + (\phi)_{i-n} & i = n+1, \ldots, 2n \end{cases} \tag{4.2}$$

where:
ϕ is the i_{th} column of $\sqrt{(n+\lambda)\mathbf{P_{k-1}}}$,
$\mathbf{P_{k-1}}$ is the covariance estimate from the previous iteration,
$\mathbf{\bar{X}_{k-1}}$ is the state estimate from the previous iteration, and
λ is the scaling parameter.

$2n+1$ sigma points are obtained from Equation 4.2 that represents the posterior mean and covariance of the state vector. Likewise, the observation vector is propagated using the propagated state vector, with the addition of measurement (observation) noise covariance:

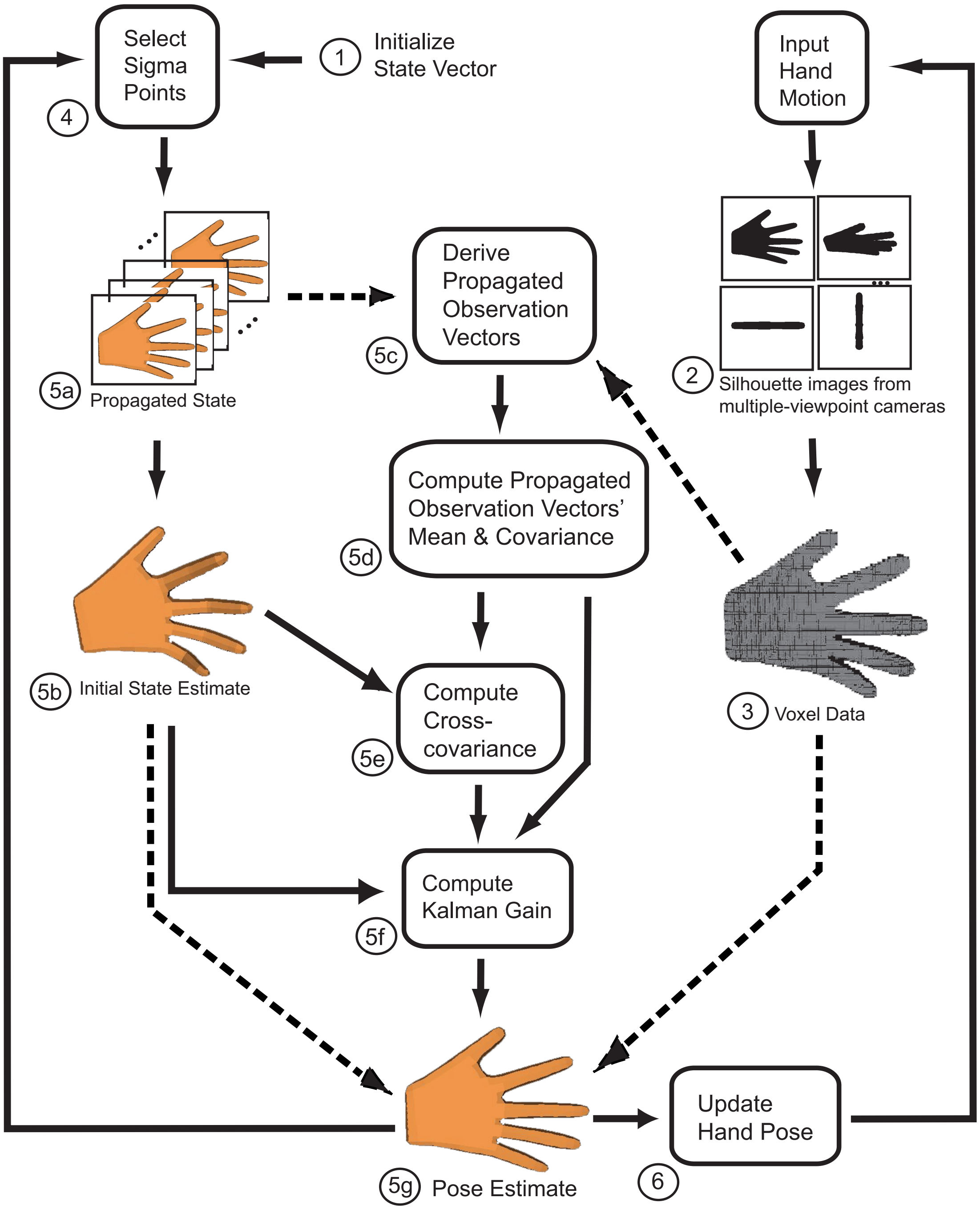


Figure 4.4: Details of the hand pose estimation using UKF. Dashed lines indicate comparison of models in order to obtain error measurements.

$$\mathbf{\hat{Y}_k} = h(\mathbf{\hat{X}_k}, \mathbf{S_k}) \longrightarrow \mathbf{\hat{Y}_k} \approx \left\{\mathbf{\hat{Y}_k^0}, \mathbf{\hat{Y}_k^1}, \mathbf{\hat{Y}_k^2}, \ldots, \mathbf{\hat{Y}_k^{2N}}\right\} \tag{4.3}$$

where:
h describes the nonlinear observation function,
$\mathbf{\hat{Y}_k}$ is the propagated observation vector,
$\mathbf{\hat{X}_k}$ is the propagated state vector, and
$\mathbf{S_k}$ is the measurement noise covariance.

Figure 4.4 illustrates the process of using UKF to estimate the skeletal pose of the hand using the voxel data as input and is explained as follows:

1. Set the state vector to $\mathbf{\bar{X}_{k-1}}$ and the state covariance to $\mathbf{\bar{P}_{k-1}}$. At initialization, set the state to zero ($\mathbf{X_0}$) and the state covariance to some value ($\mathbf{P_0}$).

2. Convert the color image inputs from the multiple cameras to silhouettes.

3. Using shape-from-silhouette technique, convert the silhouette images to voxel data.

4. Select the sigma points using $\mathbf{\bar{X}_{k-1}}$ and $\mathbf{P_{k-1}}$.

5. Estimate the hand pose using UKF:

   (a) Apply Equation 4.1, the state dynamics equation, to the sigma points $\mathbf{X_k^i}$. This gives the propagated state vectors $\mathbf{\hat{X}_k^i}$, illustrated as variations of hand poses.

   (b) Calculate the mean value of the propagated state vectors $\mathbf{\bar{\hat{X}}_k}$ and its covariance $\mathbf{\hat{P}_k}$. The $\mathbf{\bar{\hat{X}}_k}$ is the filter's initial state estimate.

   (c) Propagate the observation vector $\mathbf{\hat{Y}_k^i}$ by computing the error between the propagated state $\mathbf{\hat{X}_k^i}$ and the voxel data.

   (d) Calculate the mean value of the propagated observation vectors, $\mathbf{\bar{\hat{Y}}_k}$, and its covariance $\mathbf{\hat{P}_{yy_k}}$.

   (e) Calculate the cross covariance $\mathbf{P_{xy_k}}$.

   (f) Compute Kalman gain $\mathbf{K_k}$.

(g) Compute state estimate $\bar{\mathbf{X}}_{\mathbf{k}}$ and its covariance $\mathbf{P}_{\mathbf{k}}$. The hand pose estimate is defined by $\bar{\mathbf{X}}_{\mathbf{k}}$.

6. Update the hand pose. $\bar{\mathbf{X}}_{\mathbf{k}}$ and $\mathbf{P}_{\mathbf{k}}$ become the next iteration's $\bar{\mathbf{X}}_{\mathbf{k-1}}$ and $\mathbf{P}_{\mathbf{k-1}}$, respectively.

7. Repeat from Step 2 for the next iteration.

4.4.2 The State Dynamics and Composition of the State Vector

A key factor in using a predictive filter is using the correct state dynamics (Equation 4.1); for the hand pose estimation, a second order dynamics or constant acceleration model is used. It captures the nature of the hand motion better than a constant velocity model does, as verified in [56] and as implemented in [20][24].

In Equation (4.4), $\mathbf{X}_{\mathbf{k}}$ is the state vector at time $\mathbf{k}$, Δt is the time interval between frames, $\mathbf{V}_{\mathbf{k}}$ is the noise covariance of the state vector, and $\mathbf{I}$ is the identity matrix. The state noise covariance accounts for all the disturbances not accounted for by the dynamics; it was determined heuristically in the experiments. The uncertainties of the dynamics are modeled to be independent for the position, velocity, and acceleration components.

$$\mathbf{X}_{\mathbf{k}} = \begin{bmatrix} \mathbf{I} & \Delta t\mathbf{I} & \frac{1}{2}\Delta t^2\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \Delta t\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} [\mathbf{X}_{\mathbf{k-1}}] + [\mathbf{V}_{\mathbf{k}}] \tag{4.4}$$

The state vector $\mathbf{X}$ is composed of both global (rotation and translation) and local (finger joint angles) pose parameters and their respective first and second order derivatives (velocity and acceleration):

$$\mathbf{X} = [\mathbf{X}_{\mathbf{global}}\ \mathbf{X}_{\mathbf{local}}]^T \tag{4.5}$$

The global parameter is composed of the wrist's position in 3D: (ρ_x, ρ_y, ρ_z) and the palm orientation or roll, pitch, and yaw: (τ_r, τ_p, τ_y).

$$\mathbf{X}_{\mathbf{global}} = [\rho_x, \rho_y, \rho_z, \tau_r, \tau_p, \tau_y, \dot{\rho}_x, \dot{\rho}_y, \dot{\rho}_z, \dot{\tau}_r, \dot{\tau}_p, \dot{\tau}_y, \ddot{\rho}_x, \ddot{\rho}_y, \ddot{\rho}_z, \ddot{\tau}_r, \ddot{\tau}_p, \ddot{\tau}_y] \tag{4.6}$$

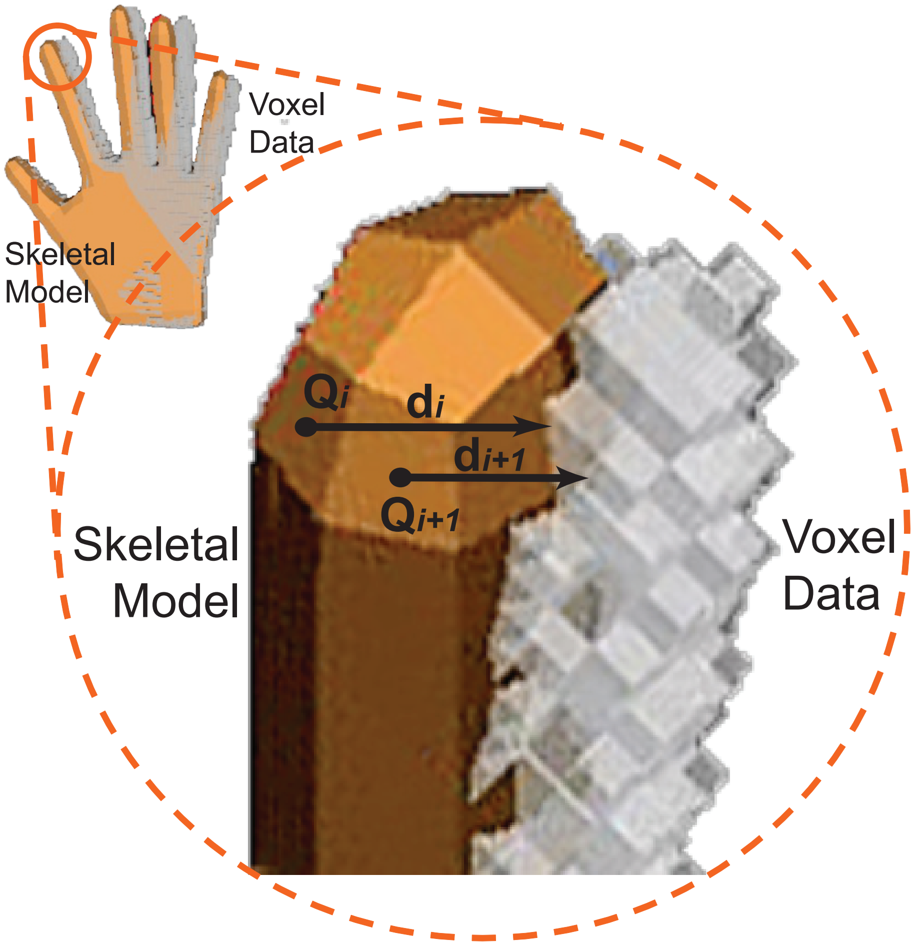


Figure 4.5: The calculation of geometric error between the skeletal model's surface structure and the voxel data.

The local parameter, the finger joint angles, is represented by θ_n, where n is the number of joint angles.

$$\mathbf{X_{local}} = \left[\theta_1, \theta_2, \ldots, \theta_n, \dot{\theta}_1, \dot{\theta}_2, \ldots, \dot{\theta}_n, \ddot{\theta}_1, \ddot{\theta}_2, \ldots, \ddot{\theta}_n\right] \tag{4.7}$$

4.4.3 The Observation Function and Composition of the Observation Vector

The observation function describes the non-linear process of obtaining the observation vector given a hand pose configuration. The voxel data of the hand, obtained by applying shape-from-silhouette technique on the camera images, is used to derive the observation vector. After modifying the hand model's pose to reflect the state vector, the error between the voxel data and the hand model is computed. The observation vector encodes the error between the observed hand pose (voxel data) and the hand model.

The observation vector is composed of geometric distance measurements between the voxel data and the hand model. Given the voxel data V and the hand model composed of i quadric surfaces, Q, the distance is computed by checking whether each quadric Q_i of the surface structure is located inside or outside of the voxel data. If it is outside, the Manhattan distance d_i between the center of the quadric and the nearest voxel is measured. If it is inside, d_i is set to zero. Figure 4.5 illustrates the process.

The distance measurements are then stacked to form the observation vector $\mathbf{Y}$:

$$\mathbf{Y} = [\ldots, d_{i-1}, d_i, d_{i+1}, \ldots]^T \tag{4.8}$$

To minimize computation time, the size of the observation vector is adjusted. In the experiments using the uncalibrated model, only 140 quadrics out of a total of 744 that make up the hand model are sampled. When the calibrated model is used, 400 voxels out of around 65000 are sampled. Additionally, at every time step, distance measurements are done on the same set of quadrics or voxels.

A zero error between the model and the observation (i.e., $\mathbf{Y} = 0$) implies that the hand model is completely inside the voxel data. Consequently, finger motion would cause the hand model to move away from the voxel, thereby Y would have some values.

4.5. Experimental Results and Discussion

4.5.1 Results Using Uncalibrated Model

For all the experiments using the uncalibrated model, the voxel data has a resolution of 2x2x2[mm] per octant. A total of 15 hand pose parameters (3 global and 12 local) were estimated. The global parameters are the roll, pitch, and yaw. The

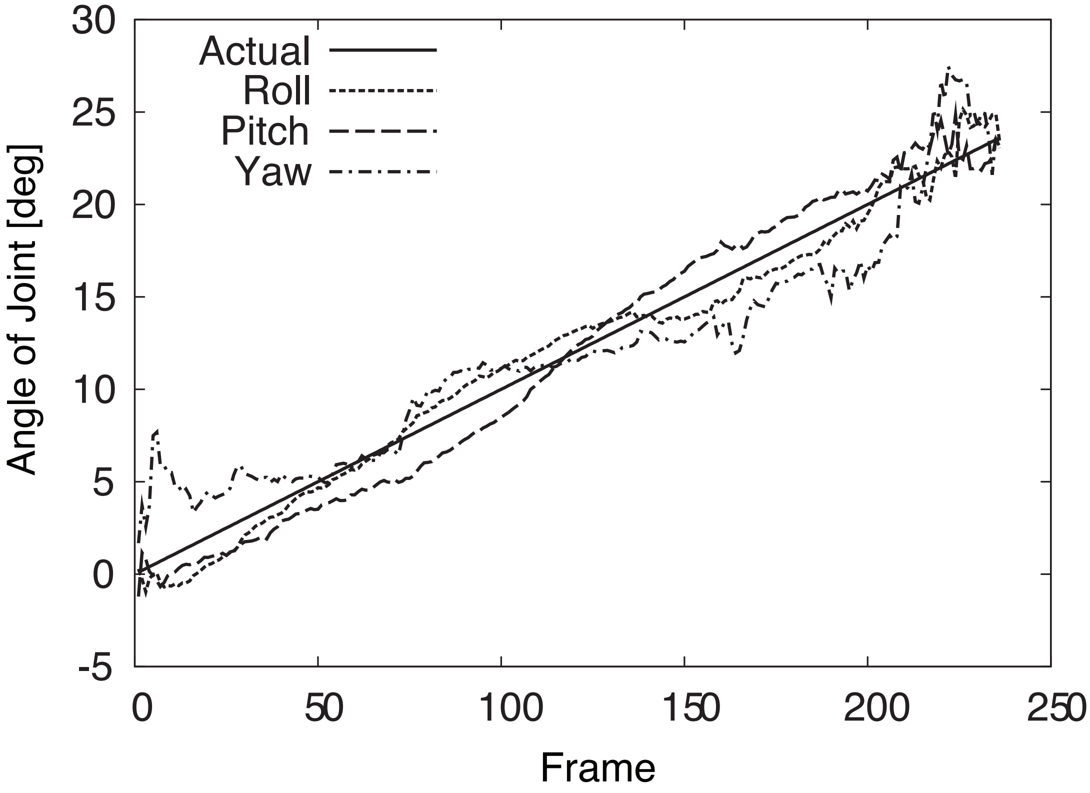


Figure 4.6: Global pose estimation result when the fingers are closing simultaneously. Solid line is the ground truth values (actual); broken lines are the estimate (roll, pitch, yaw).

local parameters are the the 2 DOFs of the MCP and 1 DOF of the PIP of all the fingers except the thumb's. The DIP's value was obtained using Equation 2.1, raising the total DOF estimated to 19.

The proposed method was tested on several hand motions. First, various hand motion data were obtained using a dataglove. These data, considered as the ground truth for all the experiments, were then fed to a motion generator to create virtual versions of the motions. These virtual motions were then used as input to the pose estimation system and then tracked. Proper initialization, i.e. alignment, of the hand model and the voxel data is necessary for filter convergence. The use of simulated motion eliminated this issue.

Figures 4.6 (Motion A), 4.7 (Motion B), and 4.8 (Motion C) show a hand motion that has been tracked successfully. Motion A is that of a hand whose wrist is rotating and twisting, while the fingers (with the exception of the thumb) are simultaneously closing slowly. The wrist's roll, pitch, and yaw (Figure 4.6) and the four fingers' PIP (1 DOF) and MCP (2 DOFs) were estimated with good accuracy. Figure 4.7 shows only the MCP's expansion-flexion data (left column) and the PIP (right column).

For Figure 4.6 and Figure 4.7, the black solid line is the ground truth value while the dotted and dashed lines are the estimate values. For all the fingers, the filter initially shows estimation errors by as much as 10 degrees, although it eventually converges to the desired value. The filter also gets lost but readjusts to get back on track. This can be seen as a noisy estimation in the pinkie's MCP joint estimation (Figure 4.7 left side, top graph). Range constraints were implemented on the finger motion to ensure that awkward poses, such as fingers bending backward too much, do not happen. This can be seen as a plateau on the pinkie's PIP estimation graph (Figure 4.7 right side, top graph).

Snapshots of the motion described above are shown in Figure 4.8. The top row is the virtually-generated motion and the bottom row is the result of the pose estimation. The numbers above each column of image correspond to the points in Figure 4.7 when the images were taken. The local motion manifests in the images as the closing and opening of the fingers, while the global motion shows as the twisting of the wrist and palm.

Two more motions were tested to demonstrate the flexibility of the system. Snapshots of the estimation results are shown in Figure 4.9 (Motion B) and Figure 4.10 (Motion C). For both motions, the wrist is rotating and twisting due to roll, pitch, and yaw motions. For Motion B, the hand is moving two fingers at a time. For Motion C, the fingers are successively bending towards the palm one by one, starting from the pinkie toward the index finger and then opening in the reverse order.

To compare the accuracy of the estimation results, Figure 4.11 shows the average of absolute errors for all the joints estimated. The absolute errors range from 0.20 to 3.40 degrees per joint for every iteration. However, the actual change of angle per iteration of any joint, based on the ground truth data, is only less than 1 degree. The converging behavior is noticeable in the graphs of Figure 4.6

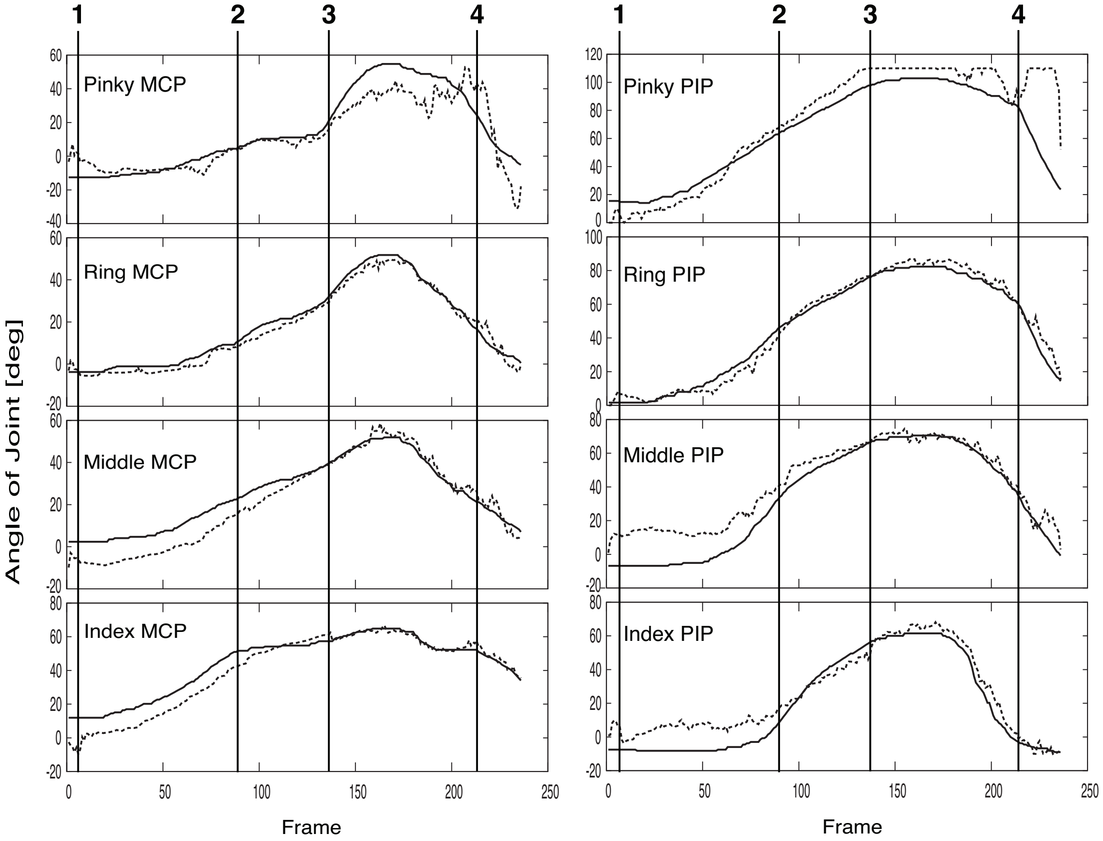


Figure 4.7: MCP and PIP estimation results for fingers closing simultaneously while the wrist is rotating. Solid lines are the ground truth values; the dotted lines are the pose estimation result. The numbered vertical lines show when the snapshots in Figure 4.8 were taken.

and 4.7 but imperceptible in the snapshots of Figure 4.8.

Comparison with Model-fitting Approach

Furthermore, estimation results using the UKF were compared with the original model-fitting approach by Ueda *et al.* [22]. Figure 4.12 establishes the robustness of using the UKF against using virtual force based model-fitting. The figure shows the estimation result of both methods for the Index finger's PIP joint. Both methods try to converge to the true value, but a closer look shows that the model-fitting has more difficulty in doing so. Between frames 100 to 200, the Index PIP is expanding and flexing (i.e., bending and stretching). The filter's estimation results fluctuate as it tries to converge to the true value yet manages to recover from the fluctuations. On the other hand, it takes some time for the model-fitting approach to recover from its over-estimation and overshoots its estimates. In short, using UKF showed better error recovery than the model-fitting method.

4.5.2 Results Using Calibrated Model

Pose estimation using the UKF was also tested with the calibrated model. Similar set of parameters as with the uncalibrated model were used. For example, the state vector was also set to 45, composed of the local and global parameters and their corresponding velocity and acceleration. The global parameters are the roll, pitch, and yaw. The local parameters are the the 2 DOFs of the MCP and 1 DOF of the PIP of all the fingers except the thumb's. The observation vector was also composed of 3D distance measurements between the model and the observation.

The input motion used was first obtained using a dataglove. Then the measurement values were fed into a hand simulator to generate the input images for four cameras. Each octant or voxel measures 2x2x2[mm].

One main difference when using the calibrated model is the size of the observation vector, which was increased to 400. This was necessary to reflect the increase in the number of sampling points on the skin of the calibrated model.

The result of the estimation is shown in Figure 4.13 for the global, Figure 4.14, 4.15, 4.16, and 4.17 for the local parameters. For all the figures, the black solid line represents the actual value of the parameters, as measured by the cyberglove, while the colored lines represent the estimation result. For Figure 4.13, the actual values for the global parameters were not obtained through dataglove but generated from within the hand simulator.

Unlike when using the uncalibrated model, the calibrated model yielded noisier pose estimation result. The fluctuation in the estimation result shows the attempts of the filter to converge to the actual value. Figure 4.18 shows the estimation result for only the global parameters. In this case, the size of the observation vector is only 9 (roll, pitch, and yaw, and their first and second order derivatives). However, similar fluctuating behavior like in Figure 4.13 was observed.

4.5.3 Implementation Issues

Initialization and Filter Tuning

Filter fine tuning and proper parameter initialization are important tasks when incorporating a predictive filter into a motion tracking solution. As mentioned above, the state vector is initialized to zero ($\mathbf{X_0}$) at the initial step. The initial pose is when the palm is flat open and the fingers are extending away from the palm.

The state covariance matrix's diagonal is set to some value ($\mathbf{P_0}$) at initialization. For the experiments, the value for $\mathbf{P_0}$ was determined heuristically.

The fine-tuning parameter λ, composed of three sub-parameters α, κ, and β, was also determined heuristically. The details of this parameter is discussed in the Appendix: Equation D.3. Van der Merwe [59] recommends setting α to a small value between 1 and 1×10^{-4}, κ to $(3 - n)$, and β to 2 for a Gaussian distribution. Likewise, the selection of the noise covariances R_k and S_k is also critical.

The other important factor in tuning UKF is the noise covariance of the state (Equation 4.1) and the observation (Equation 4.3) vectors. The stability and convergence of the filter depend on a good choice of covariances [55]. The noise covariances for both the state and observation vectors used in the experiments were determined heuristically. Table 4.1 lists the noise covariances used for the

Table 4.1: Covariance values used for the state vector using the uncalibrated model.

State Parameter	Covariance Value
θ	0.1
$\dot{\theta}$	0.01
$\ddot{\theta}$	0.001

Table 4.2: Covariance values used for the observation vector using the uncalibrated model.

Hand Motion	Covariance Value
Motion A (Figure 4.8)	0.001
Motion B (Figure 4.9)	0.1
Motion C (figFigure 4.10)	0.1

state vector when uncalibrated model is used. Table 4.2 lists the different noise covariances for the different motions of Figure 4.8, Figure 4.9, and Figure 4.10. Table 4.3 lists the state and observation noise covariances used when the calibrated model was used.

Observation Vector

The composition of the observation vector is another major factor in UKF. In the experiments, the dimension of the observation vector's was 140 for the uncalibrated model and 400 for the calibrated model. The size of the observation vector was chosen to optimize the trade-off between accuracy and computational speed. If the observation vector is too small, there would not be enough information for the filter to process, but too big a size and the computation time increases considerably.

For both uncalibrated and calibrated model experiments, the same type of measurement data was used: 3D distance between the model and the observation (voxel) data. However, the uncalibrated model performed better than the calibrated model. The difference in the performance might be explained by two factors: sampling density and sampling location.

Table 4.4 shows the number of points on the skin of each link the calibrated model that is available for measurement. Each point corresponds to the center

Table 4.3: Covariance values used for the state vector using the calibrated model.

Parameter (global and local)	**Covariance Value**
State: θ	0.1
State: $\dot{\theta}$	0.01
State: $\ddot{\theta}$	0.001
Observation	0.1

Table 4.4: Sampling points for the uncalibrated model. The thumb has no MCP-PIP link.

	Thumb	Index	Middle	Ring	Pinkie
Wrist-CM	56	62	52	78	56
CM-MCP	34	34	36	38	30
MCP-PIP	-	32	32	32	32
PIP-DIP	28	28	28	28	20
DIP-Tip	0	0	0	0	0

of the quadric face; the number of quadrics in a model is the same for all users. Table 4.5 shows the number of points on the skin of each link of the uncalibrated model for one user. Each point corresponds to a vertex of the octant; the number of points change from user to user, depending on the result of the model calibration. Comparing the two models, the calibrated model has more surface points than the uncalibrated model. Thus, despite the almost three times increase in the observation vector, the uncalibrated model is still under sampled. This concern is closely related to the sampling location. With the sheer number of sampling points in the calibrated model, it's difficult to ensure that the measurements are taken from all around each link. If the samples are taken from a narrow region of the finger link, then it might not reflect the true state of the link.

One possible solution to this sampling problem is to ration the sampling per link. The higher the number of surface points, the higher number of samples should be taken from that finger link. It would ensure that the number of points sampled per link is proportional to the total number of available points. Moreover, sampling at fixed locations distributed all throughout the link could also help in tracking its movement.

Table 4.5: Sampling points for the calibrated model. The thumb has no MCP-PIP link.

	Thumb	Index	Middle	Ring	Pinkie
Wrist-CM	47414	899	866	1560	5154
CM-MCP	4433	7508	7645	9119	14257
MCP-PIP	-	2826	4469	6633	3449
PIP-DIP	7296	4529	3878	3227	3520
DIP-Tip	4837	5991	4288	4669	3586

The above solutions would still necessitate an increase in the total number of sampled points, i.e., increase in the size of the observation vector. This in turn could lead to slower computation time. Thus, it might be necessary to lessen the number of the surface points of the calibrated model while keeping its shape intact. For example, conversion of voxels into mesh could help minimize the number of surface points.

Computation Speed

As mentioned in above, the size of the observation and the state vector could affect the processing speed of the UKF. For the state vector, the size is largely determined by the dynamics model of the system. Since a constant acceleration dynamics was chosen, the first and second derivatives of the state variables had to be incorporated in the state vector. Fortunately, consideration of known hand constraints can help lessen the dimensions of the state vector, like the coupling constraint between the PIP and the DIP (Equation 2.1); this has lessen the state vector by nine parameters. For the observation vector, the size has to be determined based on the characteristics of the model and the measurement data.

In the experiments, the computation speed of the filter for the uncalibrated model was around 0.87 seconds for every iteration or roughly 1Hz. For the calibrated model, the UKF is much slower at around three minutes per iteration. Given that the usual frame capture speed of cameras is around 30Hz, there is a strong need to speed up the proposed method.

Optimizing the size of the state and observation vectors can result in faster computations, which can lead lead to a more stable and accurate filtering. Modifications to UKF, or its equivalent methods, to further lessen the number of sigma particles from $2n + 1$ have already been reported in the literature. For example, Julier *et al.*[57] implemented a UKF with reduced sigma particles; they used only $n + 1$ sigma particles instead of $2n + 1$. La Viola compared the performance of EKF and UKF in head tracking and found that using quaternions to encode the joint angles resulted to better estimation, even by just using EKF [58].

Utilizing the advances in hardware technology can also alleviate the low computational speed of the UKF algorithm. For example, UKF's algorithm is easily parallelizable, making it easy to adapt to the multi-core multi-threading capabilities of the current generations of processors. Parallelization is easily implementable using libraries such as Intel's OpenMP. Additionally, the Graphics Processing Units (GPU) can share the load from the processors. The GPU, tailored for graphics related calculations like pixel comparison or rendering, can also be programmed to do the repetitive computation of UKF like the state and observation propagation steps.

4.6. Conclusion

To allow natural hand motion as input, Unscented Kalman Filter was used to estimate the global and local poses of the hand simultaneously. The UKF minimizes error between the hand model and the voxel data and estimates the pose by propagating $2n+1$ sigma particles. Using the uncalibrated model, estimation results of up to15 parameters (3 global and 12 local) in different motions were shown. UKF also showed better error recovery than the model-fitting pose estimation technique. It was also possible to use the UKF with the calibrated model, although the filter had a harder time in converging to the actual values. Improvements are needed in order to make UKF even more robust in estimating global and local poses of the hand. Based on the differences in performance between the calibrated and the uncalibrated model, fine tuning of the filter parameters and adjustment to the observation vector to reflect the state of the hand model need enhancement. Over-all, it has been shown that the use of predictive filter can improve pose estimation in order to make the system more adaptable to different types of input motion.

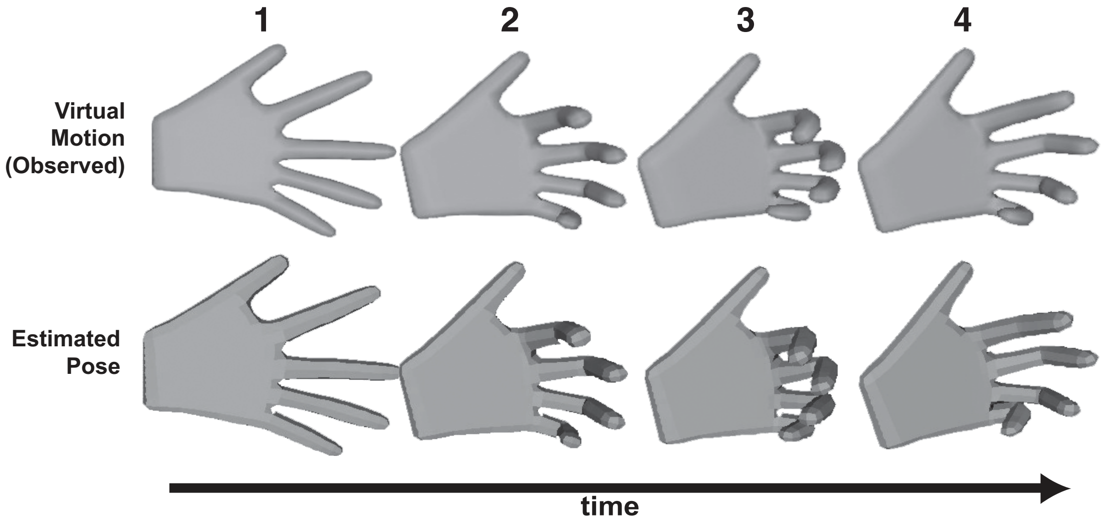


Figure 4.8: Snapshots of estimation result. The numbers above each image column correspond to the points in Figure 4.7 when the snapshots were taken. The motion is for a rotating wrist while the fingers are closing simultaneously.

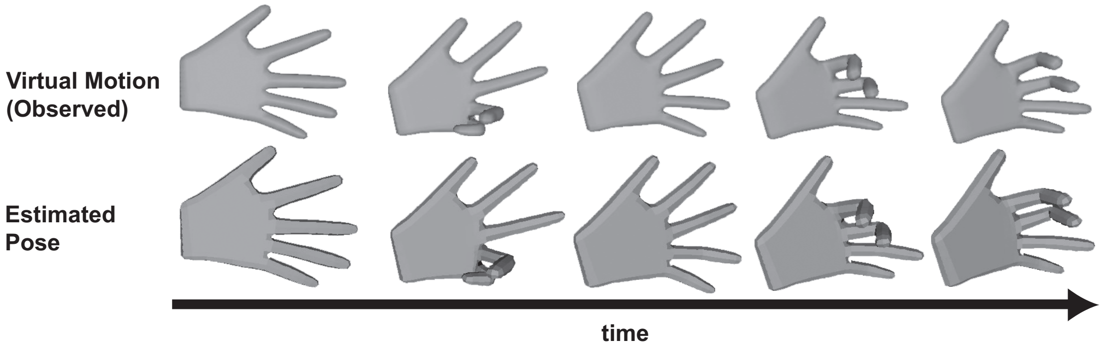


Figure 4.9: Snapshots of the observed hand motion and their corresponding estimated hand poses. The motion is that of a hand rotating while the fingers are closing two at a time.

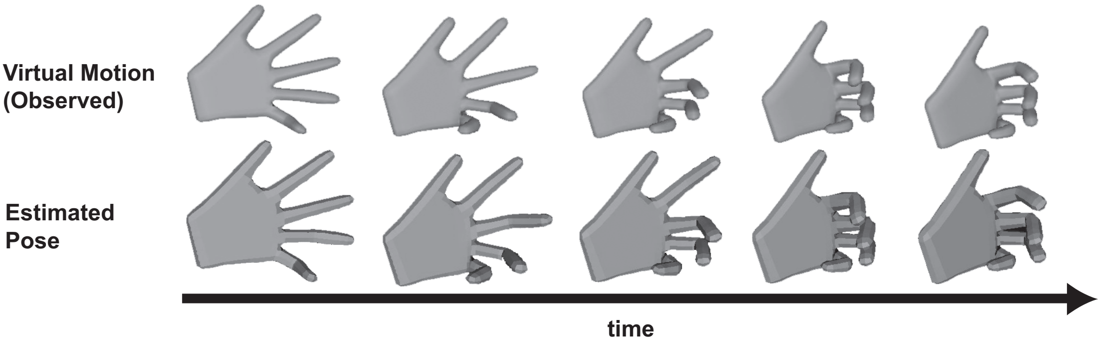


Figure 4.10: Snapshots of the observed hand motion and their corresponding estimated hand poses. The motion is that of a hand rotating while the fingers are closing one at a time starting from the pinkie going to the index.

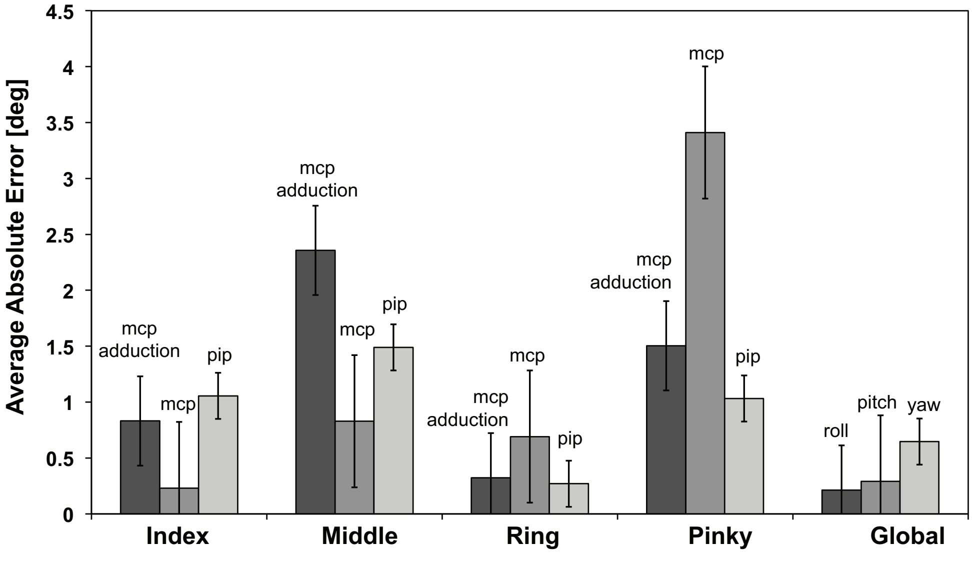


Figure 4.11: Average absolute error of each DOF. The motion is that of a hand rotating while the fingers are closing simultaneously.

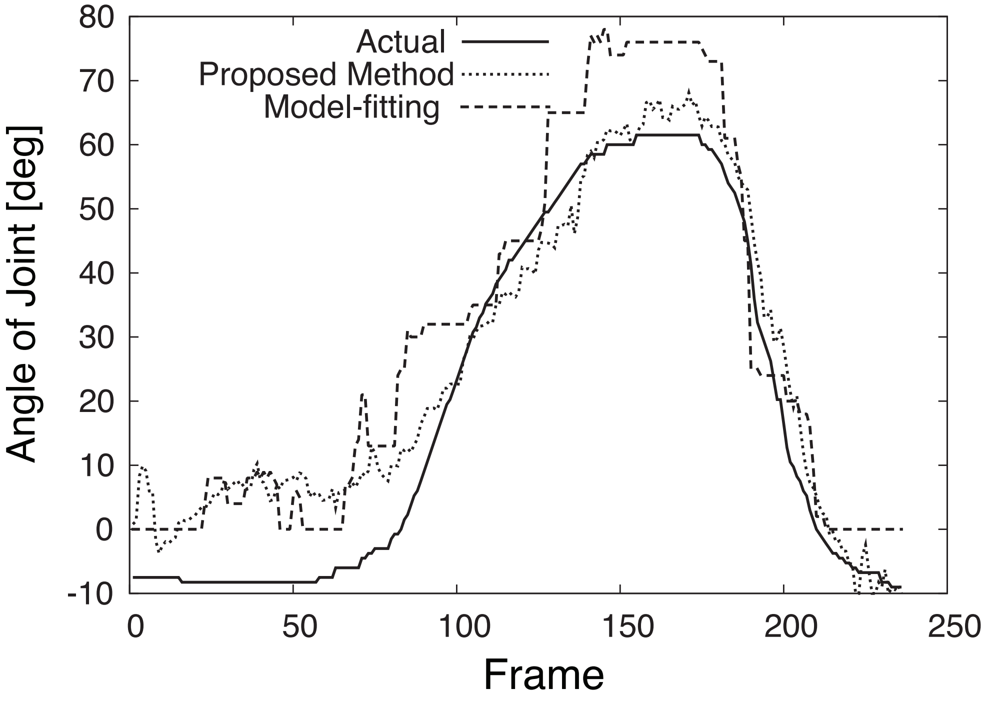


Figure 4.12: Comparison of Index PIP estimation results of the original model-fitting approach and the proposed method.

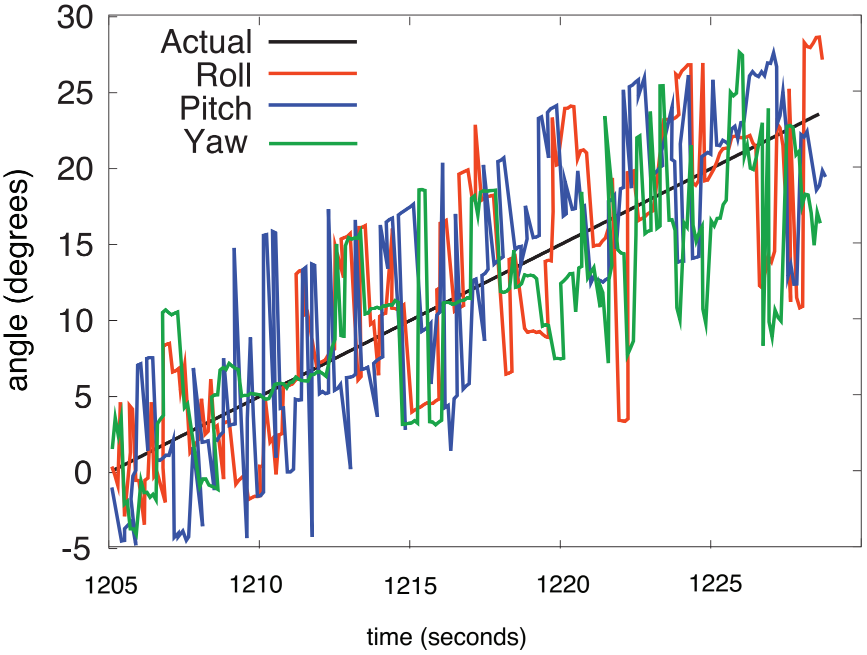


Figure 4.13: Global pose estimation result using the calibrated model. The colored lines are the pose estimation result for the roll, pitch, and yaw; the black line is the actual value. The motion is that of a hand rotating while the fingers are closing simultaneously.

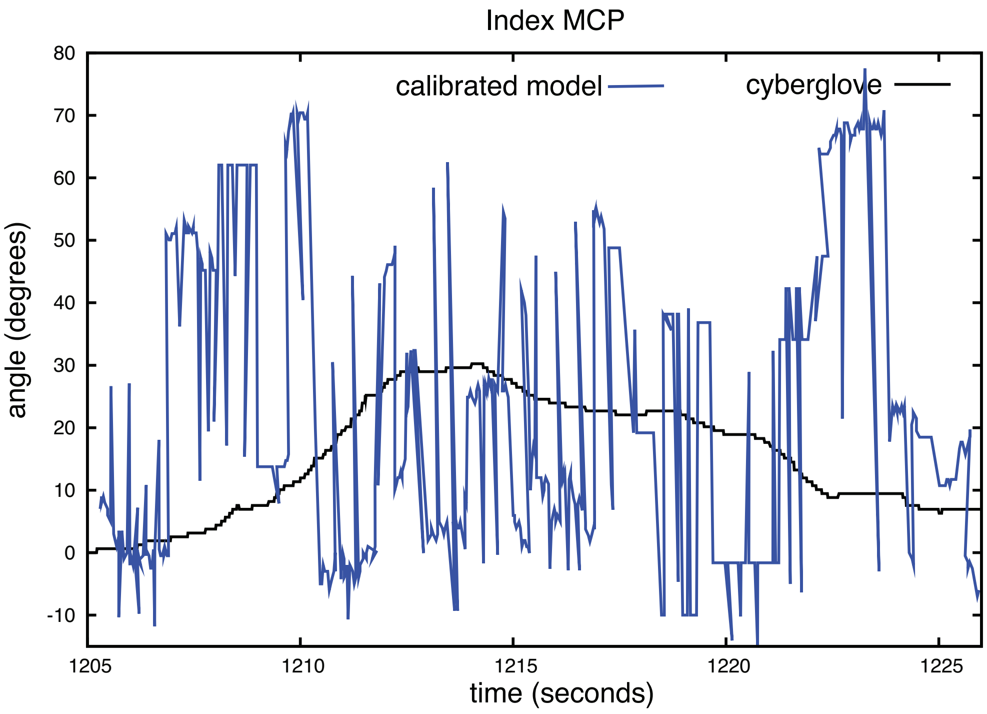


Figure 4.14: Local pose (Index MCP) estimation result using the calibrated model. The colored lines are the pose estimation result for the roll, pitch, and yaw; the black line is the actual value. The motion is that of a hand rotating while the fingers are closing simultaneously.

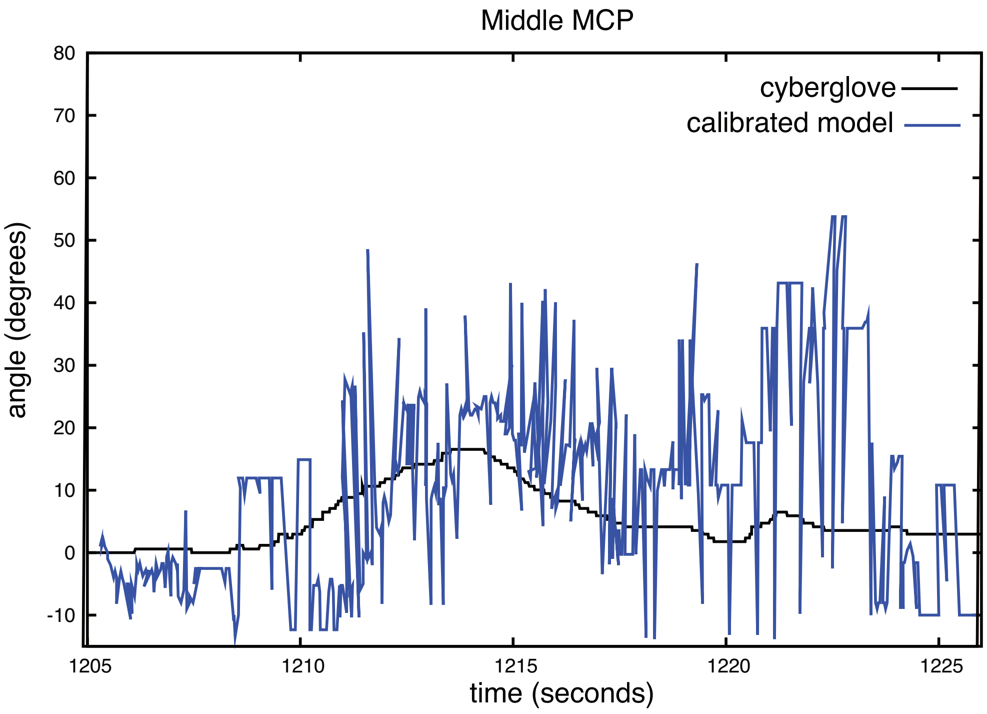


Figure 4.15: Local pose (Middle MCP) estimation result using the calibrated model. The colored lines are the pose estimation result for the roll, pitch, and yaw; the black line is the actual value. The motion is that of a hand rotating while the fingers are closing simultaneously.

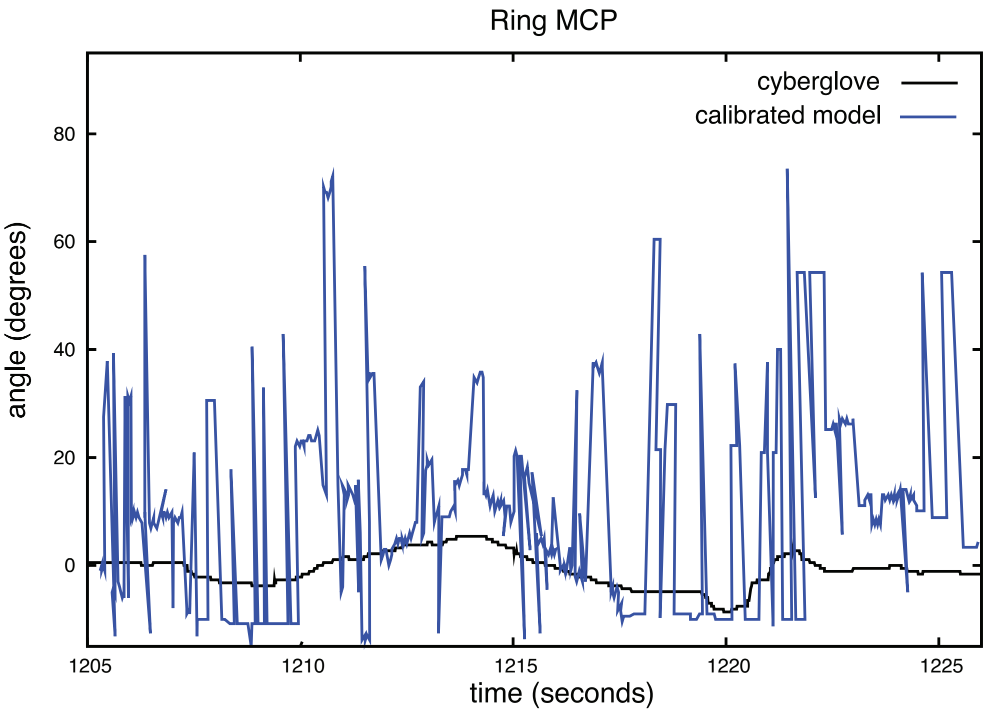


Figure 4.16: Local pose (Ring MCP) estimation result using the calibrated model. The colored lines are the pose estimation result for the roll, pitch, and yaw; the black line is the actual value. The motion is that of a hand rotating while the fingers are closing simultaneously.

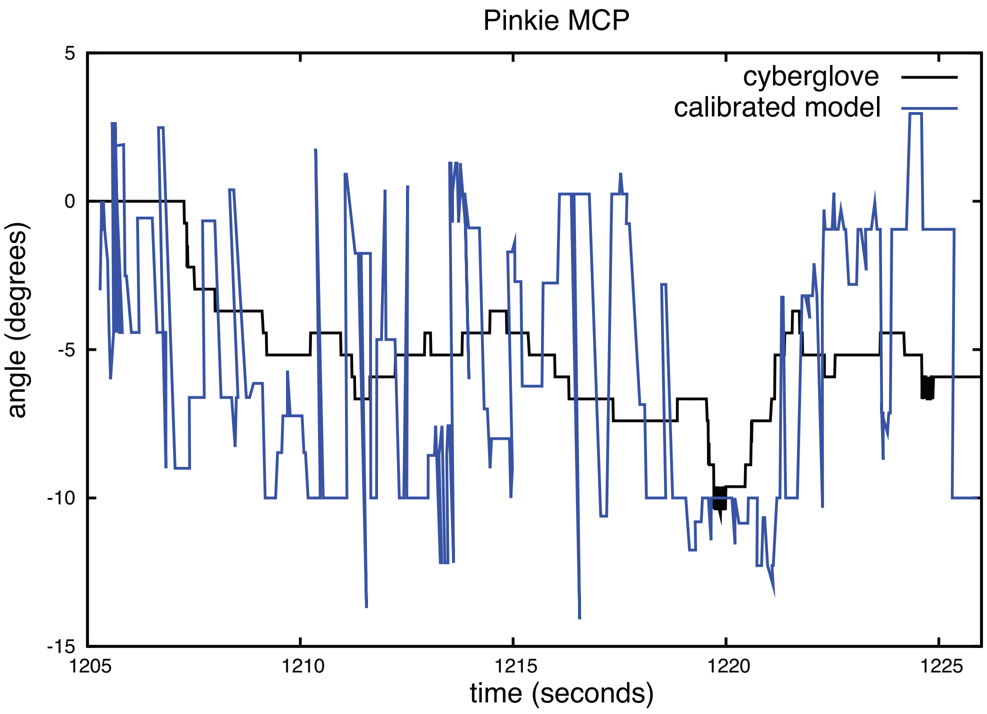


Figure 4.17: Local pose (Pinkie MCP) estimation result using the calibrated model. The colored lines are the pose estimation result for the roll, pitch, and yaw; the black line is the actual value. The motion is that of a hand rotating while the fingers are closing simultaneously.

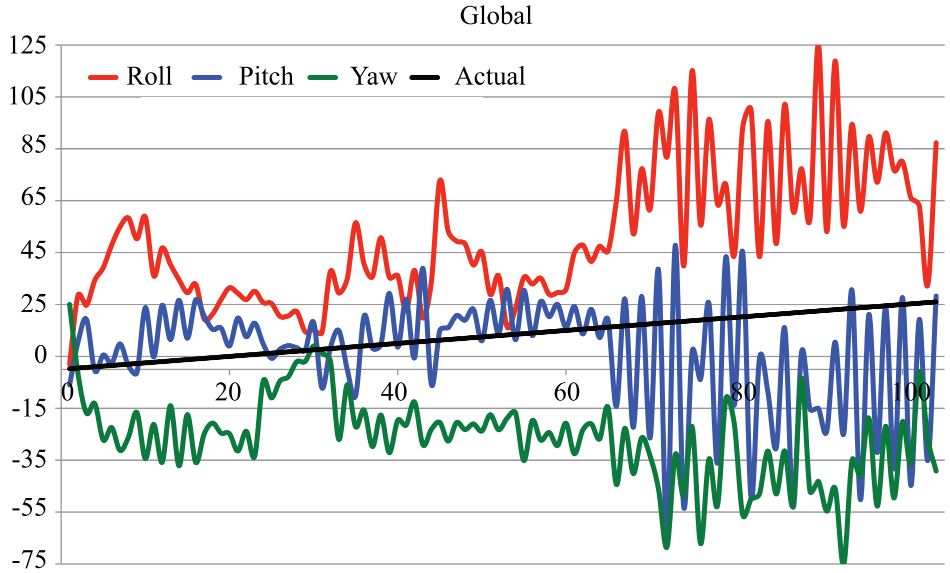


Figure 4.18: Global pose estimation result (only roll, pitch, and yaw) using the calibrated model. The colored lines are the pose estimation result for the roll, pitch, and yaw; the black line is the actual value. The motion is that of a hand rotating at its wrist.

Chapter 5

Conclusion

> What the caterpillar calls the end of the world, the master calls a butterfly.
>
> Richard Bach

Vital in making interaction between humans and robots or computers more natural is the development of a robust hand pose estimation system. An approach that permits unencumbered hand motion is vision-based and model-based. This thesis aims to contribute in this field of knowledge by addressing three main issues usually looked over among the state-of-the-art in vision-based model-based hand pose estimation research: model calibration, flexibility, and quantitative evaluation. As a solution to the raised issues, model calibration using multiple cameras and application of Unscented Kalman Filter in hand pose estimation were proposed, evaluated and discussed in this thesis.

5.1. Hand Model Individualization

In Chapter 3, the aim was to calibrate the hand model in order to allow any person, regardless of age or gender, to use the system. The model calibration was achieved by a two step process. First, voxel data of the hand was used as the skin model since the voxel data reflects the shape of the actual user hand. Then using a calibration motion, the finger link lengths, which represent the user's hand bones, were estimated by generating possible link lengths combination. The optimum combination was then chosen among the candidates using a simple search method.

To confirm the effectiveness of the approach, the hand pose estimation system was run using the calibrated model obtained. Estimation results from various users with different hand characteristics, regardless of gender and age, showed the feasibility of the proposed approach.

Calibrating the model also enabled comparison between conventional pose measurement system like dataglove and the proposed approach; it was also compared against the estimation system that uses uncalibrated model. Motions of a hand wearing a dataglove was captured and used as input to the system. Using the dataglove output as benchmark, results verified that compared to the uncalibrated model, the calibrated model estimation results showed better similarity to the dataglove measurements. To quantify the improvement when using the calibrated model, silhouettes of the estimation results were obtained and compared to the silhouettes of the input image. The calibrated model showed improvement of up to 30% in shape similarity to the input image when compared to the uncalibrated model.

In this chapter, the hypothesis that improving the hand model improves hand pose estimation was verified. Two things have been achieved here. First, model calibration was enabled that allowed users of varying hand characteristics, from child to adult, to use the hand pose estimation system. Second, a quantitative comparison of the proposed performance was attained. The encouraging results imply that it is possible to make a vision-based hand pose estimation system applicable to various users.

5.2. Predictive Filtering in Hand Pose Estimation

In Chapter 4, flexibility in pose estimation was addressed by making the system more robust against errors and enabling it to simultaneously estimate the global and local pose parameters of the hand. This was accomplished by incorporating a Bayesian-based filter, Unscented Kalman Filter, into the pose estimation design. UKF was chosen because of its ability to perform well for non-linear systems, especially since the hand motion and the relationship between the observation and the hand pose are non-linear in nature. The necessary steps in using the UKF to track the hand motion and estimate the hand pose was discussed in detail.

The system was evaluated through virtually generated hand motions using both calibrated and uncalibrated models. Dataglove measurements were used in the creation of the virtual motions, providing a baseline for comparison. Results using the uncalibrated model confirmed the improvement in error recovery and ability of the system to estimate both global and local pose at the same time. Using UKF with the calibrated model was also possible, although estimation results were not as accurate as with the uncalibrated model. This shows that further adjustments when using the calibrated model are necessary to fully realize the advantages of employing UKF.

Treating hand motion as a non-linear system and using a predictive filter can improve the pose estimation results. What has been achieved in this chapter is the creation of a system that is flexible enough to accommodate hand motion that changes at both local and global levels. Results show that global and local pose parameters can be estimated simultaneously. A complete hand pose estimation system would truly allow users to move in any which way in 3D, enriching their interaction with computers and robots alike. With this work, a hand pose estimation system that allows unhindered motion has been proven achievable.

5.3. Future Work

With the gains realized in Chapter 3 and Chapter 4, a vision-based model-based hand pose estimation system can be more universal in appeal and application by allowing various users to use the system by using their hands as input. It is hoped that this thesis lays down the ground work for further exploration in enhancing hand pose estimation for HCI and HRI.

This research work also opens up different avenues that can be pursued to further enhance the system. For the model calibration, a faster way of searching through the search space for optimum link lengths is an area to explore. Making the calibration online and real-time is also a worthwhile topic.

In using UKF, some limitations like long term stability and initialization of the hand model have been observed. Works that address this, either by using a new class of filters, or by modifying the UKF to suit the hand problem domain are called for. To improve the performance of the calibrated model at par with the uncalibrated model, modification to the observation vector, and possibly with the model itself, is necessary. A method to minimize the surface points of the calibrated hand model while retaining its shape can help the system become more robust and faster. Currently, the heavy computational requirements of UKF does not allow the system to be tested in real-time; speeding up the system is another future task.

Acknowledgements

ad maiorem + Dei gloriam

St. Ignatius of Loyola

First of all, I would like to thank my professor and thesis adviser, Professor Tsukasa Ogasawara. He has welcomed me in the Robotics Laboratory and helped me in fulfilling my dream. His unassuming ways of managing my research has helped me a lot in growing as a researcher and as a person. With his guidance and generosity, I was able to accomplish this thesis. Without him, this thesis would not be a success.

A special thanks to my other thesis adviser, Professor Etsuko Ueda, now in Nara National College of Technology. I joined the lab in 2004, and she was the Assistant Professor then who immediately took me under her wings and introduced me to the wonderful topic of hand pose estimation. She has been a constant guide in my journey through this research and taught me different ways of looking at my work and approaching the problems I encountered. She has been an adviser, in research and in life, and an example of perseverance and dedication to work.

I would also like to thank the other members of the Robotics Lab staff, both past and present: former Associate Professor Yoshio Matsumoto (currently in NICT and my former thesis advisor) and former Assistant Professor Jun Ueda (currently in Georgia Institute of Technology). Their insightful comments and advice made a lot of difference in my work.

To Associate Professor Jun Takamatsu, also my thesis adviser, and Assistant Professor Kentaro Takemura. They pushed me to greater heights and gave me much needed advice on how to go about my research. Their forthright and discerning guidance steered me toward the right direction and helped me clear the path towards the completion of this thesis.

Besides my advisors, I would also like to thank the rest of my thesis committee: Professor Naokazu Yokoya, Thank you very much for reviewing my thesis and for the invaluable guidance.

I would also like to thank all the members of the Robotics Lab, past and present, many of whom have become like a family to me. Special mention to Assistant Professor Yuichi Kurita, Junichi Ido, Masanao Koeda, Abdel Aziz Khiat, Masahiro Kondo, Tsuyoshi Suenaga (who was also my tutor when I first arrived in the lab), Ding Ming, Akihiko Yamaguchi, Atsutoshi Ikeda, Mai Matsuo, Daisuke Kuraki, Kazuki Yamamoto.

Let me also thank the administrative staffs in the lab who has been patient with me and helped me with the bureaucratic side of being a student: Megumi Kanaoka, Miyuki Yamaguchi, and Michiko Owaki.

I would like to thank all my friends here in Japan and back home in the Philippines for joining me in my life travels all throughout these years. Kudos and thanks to the first group of Filipinos who has passed through the hallways of NAIST (Randy Gomez, Joanne Dy-Yu, and Edison Yu); And to the current ones, thanks for making life better for each one of us. To my friends from Tokyo, especially members of Hiyas ng Pinas, to my friends back home who never stopped believing in me, Ruel Justiza, Joub Miradora, and Paolo dela Rama, my endless thanks.

To my parents, Lamberto and Marilou, who, despite wondering why I have to stay in Japan for so long just to study, still believed and supported me; to my sisters and brother, Berlyn, Zhee, Jenette, Cecille, and RJ, who have been my constant source of joy and inspiration.

And to my loving and caring wife: Zarah, the wind beneath my wings. For being my source of strength and hope, for always being there, for reminding me of the things I love and helping me remember why I do what I do: I love you.

And to Him, where everything begins and everything ends. May this work be a testament of his infinite love, grace, and mercy.

Publication List

Journal

1. Albert Causo, Etsuko Ueda, Kentaro Takemura, Yoshio Matsumoto, Jun Takamatsu and Tsukasa Ogasawara. User-adaptable Hand Pose Estimation Technique for Human Robot Interaction. *Journal of Robotics and Mechatronics*, 21(6):739-748, December 2009.

Book Chapter

1. Albert Causo, Etsuko Ueda, Kentaro Takemura, Yoshio Matsumoto, Jun Takamatsu and Tsukasa Ogasawara. Predictive Tracking in Vision-based Hand Pose Estimation Using Unscented Kalman Filter and Multi-viewpoint Cameras. In: *Human-Robot Interaction*, Daisuke Chugo (Ed.), ISBN: 978-953-307-051-3, Crotia: INTECH, February 2010. pp.155-170.

International Conferences

1. Albert Causo, Mai Matsuo, Etsuko Ueda, Kentaro Takemura, Yoshio Matsumoto, Jun Takamatsu and Tsukasa Ogasawara. Hand Pose Estimation using Voxel-based Individualized Hand Model. In *Proc. of The 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2009).* pages 451-456. Suntec Convention and Exhibition Center, Singapore. 14-17 July 2009.

2. Albert Causo, Mai Matsuo, Etsuko Ueda, Yoshio Matsumoto, and Tsukasa Ogasawara. Individualization of voxel-based hand model. In *Proc. of*

The 4th ACM/IEEE International Conference on Human Robot Interaction (HRI 2009). pp. 219-220. La Jolla, California, USA, March 2009.

3. Albert Causo, Etsuko Ueda, Yuichi Kurita, Yoshio Matsumoto, Tsukasa Ogasawara. Model-based Hand Pose Estimation Using Multiple Viewpoint Silhouette Images and Unscented Kalman Filter. In *Proc. of The 17th International Symposium on Robot and Human Interactive Communication (RO-MAN 2008).* pp. 291-296. Munich, Germany, August 2008.

Domestic Conferences

1. Albert Causo, Etsuko Ueda, Yoshio Matsumoto, Tsukasa Ogasawara. Simultaneous Estimation of Hand Pose Parameters using Unscented Kalman Filter. In *Proc. of The 25th Annual Conf. of the Robotics Society of Japan.* 1H24, Sep 2007.

2. Albert J. Causo, Etsuko Ueda, Yoshio Matsumoto, Tsukasa Ogasawara. Simultaneous Estimation of Hand-Pose Parameters Using Multiviewpoint Silhouette Images. In *Proc. of The 6th Annual Conf. of SICE System Integration Division of Japan (SI2005)*, 1B2-4, December 2005.

References

[1] W. Penfield and T. Rasmussen. *The cerebral cortex of man: A clinical study of localization of function.* MacMillan, New York, 1950.

[2] Chairman Thomas T. Hewett. ACM SIGCHI Curricula for Human-Computer Interaction. Technical report, ACM, New York, NY, USA, 1992.

[3] V.I. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, July 1997.

[4] J. Crowley, F. Berard, and J. Coutaz. Finger tracking as an input device for augmented reality. In *Proc. of Int. Workshop on Gesture and Face Recognition*, Zurich, June 1995.

[5] W. Freeman, D. Anderson, P. Beardsley, C. Dodge, M. Roth, C. Weissman, W. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, and K. Tanaka. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, 18(3):42–53, 1998.

[6] A. Torige and T. Kono. Human interface by recognition of human gestures with image processing: Recognition of gesture to specify moving direction. In *Proc. of IEEE Int. Workshop on Robot and Human Communication*, pages 105–110, 1992.

[7] J. Segen and S. Kumar. Shadow gesture: 3D hand pose estimation using

a single camera. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 479–485, 1999.

[8] N. Shimada, K. Kimura, Y. Shirai, and Y. Kuno. Hand-posture estimation by combining 2D appearance-based and 3D model-based approaches. In *Proc. of Int. Conf. on Pattern Recognition*, volume 3, pages 3709–3712, Barcelona, Spain, September 2000.

[9] V. Athitsos and S.J. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition*, volume 2, pages 432–439, Madison, WI, USA, June 2003.

[10] A. Utsumi and J. Ohya. Multiple-hand gesture tracking using multiple cameras. In *Proc. of the IEEE Computer Society Conference on Comp. Vis. and Pattern Recognition*, volume 1, pages 473–478, Ft. Collins, CO, USA, June 1999.

[11] Y. Wu and T. S. Huang. View-independent recognition of hand postures. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 88–94, Hilton Head Island, SC, USA, June 2000.

[12] N. Shimada, K. Kimura, and Y. Shirai. Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera. In *Proc. of IEEE ICCV Workshop on Recognition, Analysis, Tracking of Faces and Gestures in Real-Time Systems*, pages 23–30, Vancouver, Canada, July 2001.

[13] R. Rosales, S. Sclaroff, and V. Athitsos. 3D hand pose reconstruction using specialized mappings. In *Proc. of IEEE Int. Conf. on Computer Vision*, volume 1, pages 378–385, Vancouver, Canada, July 2001.

[14] Y. Wu, J.Y. Lin, and T.S. Huang. Capturing natural hand articulation. In *Proc. of the Eighth IEEE Int. Conf. on Computer Vision*, volume 2, pages 426–432, Vancouver, BC, July 2001.

[15] J. M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: An application to human hand tracking. In J.O. Eklundh, editor, *Proc. of Third European Conf. on Computer Vision*, pages 35–46, Stockholm, Sweden, May 1994.

[16] J.M. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Proc. of IEEE Workshop on Motion of Non-Rigid An Articulated Objects*, pages 16–22, Austin, TX, USA, November 1994.

[17] K. Nirei, H. Saito, M. Mochimaru, and S. Ozawa. Human hand tracking from binocular image sequences. In *Proc. of 22nd Int'l Conf. on Industrial Electronics, Control, and Instrumentation*, pages 297–302, Taipei, August 1996.

[18] T. Heap and D. Hogg. Towards 3D hand tracking using deformable model. In *Proc. of IEEE Int. Conf. Automatic Face and Gesture Recognition*, pages 140–145, Killington, VT, USA, 1996.

[19] C. Lien and C. Huang. Model-based articulated hand motion tracking for gesture recognition. *Image Vision Computing*, 16(2):121–134, feb 1998.

[20] B. Stenger, P.R.S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *Proc. of the IEEE Computer Society Conf. on Comp. Vision and Pattern Recognition*, volume 2, pages 310–315, Kauai, Hawaii, USA, December 2001.

[21] S. Lu, G. Huang, D. Samaras, and D. Metaxas. Model-based integration of visual cues for hand tracking. In *Proc. of Workshop on Motion and Video Computing*, pages 118–124, Orlando, FL, USA, December 2002.

[22] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara. A hand-pose estimation for vision-based human interfaces. *IEEE Tran. on Ind. Electron.*, 50(4):676–684, August 2003.

[23] Q. Delamarre and O. Faugeras. 3D articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, 81(3):328–357, 2001.

[24] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. *Hand Pose Estimation using Hierarchical Detection*, volume 3058 of *Lecture Notes in Computer Science: Computer Vision in Human-Computer Interaction*, pages 105–116. Springer Berlin / Heidelberg, 2004.

[25] A. Causo, E. Ueda, Y. Kurita, Y. Matsumoto, and T. Ogasawara. Model-based hand pose estimation using multiple viewpoint silhouette images and unscented kalman filter. In *Proc. of the 17th Int. Symp. on Robot and Human Interactive Communication*, pages 291–296, Munich, Germany, August 2008.

[26] A. Causo, E. Ueda, K. Takemura, Y. Matsumoto, J. Takamatsu, and T. Ogasawara. Hand pose estimation using voxel-based individualized hand model. In *Proc. of the 2009 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pages 451–456, Singapore, July 2009.

[27] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, and X. Twombly. Vision-based hand motion estimation: A review. *Comput. Vis. Image Underst.*, 108(1-2):52–73, October 2007.

[28] S. Waldherr, R. Romero, and S. Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[29] O. Rogalla, M. Ehrenmann, R. Zöllner, R. Becher, and R. Dillmann. Using gesture and speech control for commanding a robot assistant. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 454–459, Berlin, Germany, September 2002.

[30] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2422–2427, Sendai, Japan, September 2004.

[31] J. Kofman, W. Xianghai, T.J. Luu, and S. Verma. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Tran. on Ind. Electron.*, 52(5):1206–1219, 2005.

[32] I. Infantino, A. Chella, H. Džindo, and I. Macaluso. Visual control of a robotic hand. In *The IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2003)*, volume 2, pages 1266 – 1271, Las Vegas, NV, USA, October 2003.

[33] A. Chella, H. Džindo, I. Infantino, and I. Macaluso. A posture sequence learning system for an anthropomorphic robotic hand. *Robotics and Autonomous Systems*, 47(2-3):143–152, 2004.

[34] S. Thrun. Toward a framework for human-robot interaction. *Human-Computer Interaction*, 19(1):9–24, 2004.

[35] M.A. Goodrich and A.C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[36] N. Palastanga, D. Field, and R. Soames. *Anatomy and Human Movement: Structure and Function.* Elsevier Science Ltd., 4th edition, 2002.

[37] A. Hollister, W.L. Buford, L.M. Myers, D.J. Giurintano, and A. Novick. The axes of rotation of the thumb carpometacarpal joint. *Journal of Orthopaedic Research*, 10(3):454–460, 2005.

[38] J.J. Kuch and T.S. Huang. Vision-based hand modeling and tracking: A hand model. In *Proc. of Twenty-Eighth Asilomar Conference on Signal, Systems and Computers*, pages 1251–1256, November 1994.

[39] J. Lin, Y. Wu, and T.S.Huang. Modeling the constraints of human hand motion. In *IEEE Workshop on Human Motion*, pages 121–126, Washington, DC, USA, 2000.

[40] C.L. Huang and S.H. Jeng. A model-based hand gesture recognition system. *Machine Vision and Applications*, 12(5):243–258, March 2007.

[41] A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla. Learning a kinematic prior for tree-based filtering. In *Proc. of British Machine Vision Conf.*, volume 2, pages 589–598, Norwich, UK, September 2003.

[42] M. Bray, E. Koller-Meier, and L.V. Gool. Smart particle filtering for 3D hand tracking. In *Proc. of the Sixth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, page 675, 2004.

[43] N.Shimada, Y.Shirai, Y.Kuno, and J.Miura. Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraint. In *Proc. of the Third IEEE Int. Conf. on Face and Gesture Recognition*, pages 268–273, Nara, Japan, April 1998.

[44] A. Thayananthan, B. Stenger, PHS Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 127–134, Wisconsin, USA, June 2003.

[45] B. Buchholz, T.J. Armstrong, and S.A. Goldstein. Anthropometric data for describing the kinematics of the human hand. *Ergonomics*, 35(3):261–273, 1992.

[46] C.S. Chua, H. Guan, and Y.K. Ho. Model-based 3D hand posture estimation from a single 2D image. *Image and Vision computing*, 20(3):191–202, 2002.

[47] T. Kurihara and M. Miyata. Modeling deformable human hands from medical images. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 355–363, 2004.

[48] T. Rhee, J.P. Lewis, U. Neumann, and K. Nayak. Soft-tissue deformation for in vivo volume animation. In *Proc. of 15th Pacific Conf. Comp. Graphics and Applications*, pages 435–438, 2007.

[49] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.

[50] Y. Hattori, A. Nakazawa, and H. Takemura. Refinement of the shape reconstructed by visual cone intersection using fitting the standard human model. In *IPSJ SIG Notes CVIM*, volume 31, pages 147–154, 2007.

[51] Y. Azoz, L. Devi, and R. Sharma. Tracking hand dynamics in unconstrained environments. In *Proc. of the Third IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 274–279, Nara, Japan, April 1998.

[52] S.J. Julier and J.K Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proc. of Conf. on Signal Processing, Sensor Fusion, and*

Target Recognition VI, volume 3068, pages 182–193, Orlando, FL, USA, April 1997.

[53] T. Gumpp, P. Azad, K. Welke, E. Oztop, and R. Dillmann. Unconstrained real-time markerless hand tracking for humanoid interaction. In *Proc. of Sixth IEEE-RAS Int. Conf. on Humanoid Robots*, pages 88–93, Genova, Italy, December 2006.

[54] J.Y. Lin, Y. Wu, and T.S. Huang. Capturing hand motion in image sequences. In *Proc. of IEEE Workshop on Motion and Video Computing*, pages 99–104, Orlando, FL, USA, December 2002.

[55] K. Xiong, H.Y. Zhang, and Chan C.W. Performance evaluation of ukf-based nonlinear filtering. *Automatica*, 42(2):261–270, February 2006.

[56] A. Causo. Introducing unscented kalman filter to hand pose estimation. Master's thesis, Nara Institute of Science and Technology (NAIST), Ikoma City, Nara, Japan, March 2006.

[57] S.J. Julier and J.K. Uhlmann. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *Proc. of the 2003 IEEE American Control Conf.*, pages 887–892, Anchorage AK, USA, May 2002.

[58] J.J. La Viola Jr. A comparison of unscented and extended kalman filtering for estimating quaternion motion. In *Proc. of the 2003 American Control Conf.*, volume 3, pages 2435–2440, Denver, CO, USA, June 2003.

[59] E. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proc. of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symp.*, pages 153–158, Lake Louise, Alberta, Canada, October 2000.

Appendix

A. Octree and Voxel

The main data used in this thesis is the voxel and the procedure for creating the voxel is taken from the work of Szeliski [49]. After color images of an object from multiple viewpoints are taken and binarized, they are now ready to be turned into voxel.

The first step is creating an octree from the different images. The whole workspace is assumed to be represented by a cube, which can be subdivided incrementally in chunks of eight (hence, octree). Each part is called an octant.

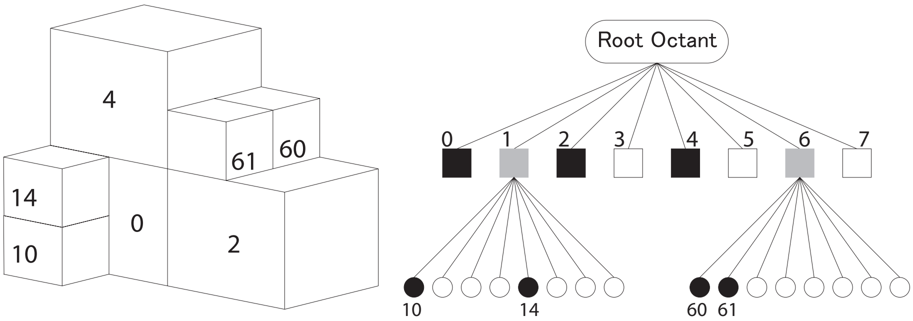


Figure A.1: The octree structure.

As shown in Figure A.1, the topmost level of the tree represents the whole workspace. When sliced into eight parts, each part belongs to the second level of the tree or the children of the root. Further subdivision of a second level octant into eight parts yields the third level octants which is now just $\frac{1}{64}$th of the root octant at level 1. Since each octant represents a volume in 3D space, it is also known as a voxel or volume pixel. The workspace's volume can be subdivided depending on the accuracy needed to represent an object [49].

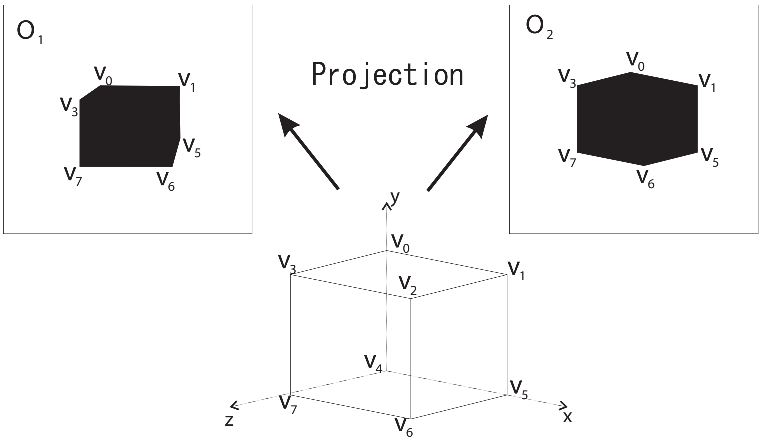


Figure A.2: Octant projection.

The next step, done hierarchically and iteratively, is carving away the octants or voxels that do not form part of the object. Each voxel is projected to 2D according to the camera parameter used in the taken image. In general, a voxel forms a hexagonal shadow, as shown in Figure A.2.

Beginning with the root octant, each voxel is projected to 2D and the vertices of the hexagonal shadow is compared to the silhouette of the binarized image. When all the vertices of the hexagon are inside the silhouette area, then the voxel is considered as part of the object's interior and is labeled BLACK. When all the vertices are outside the silhouette area, the voxel is considered as part of the background and is labeled WHITE. When only part of the voxel is inside the silhouette, then the voxel is considered as boundary case and is labeled as GRAY. Szeliski accelerated the comparison by using the four vertices of the box that bounds the hexagon [49]. The comparison cases are illustrated in Figure A.3.

The process continues until the desired level of voxel has been reached, thus, the process is hierarchical. After processing an image, the same procedure is repeated for the next image, until all images are exhausted. Thus, the process is also iterative.

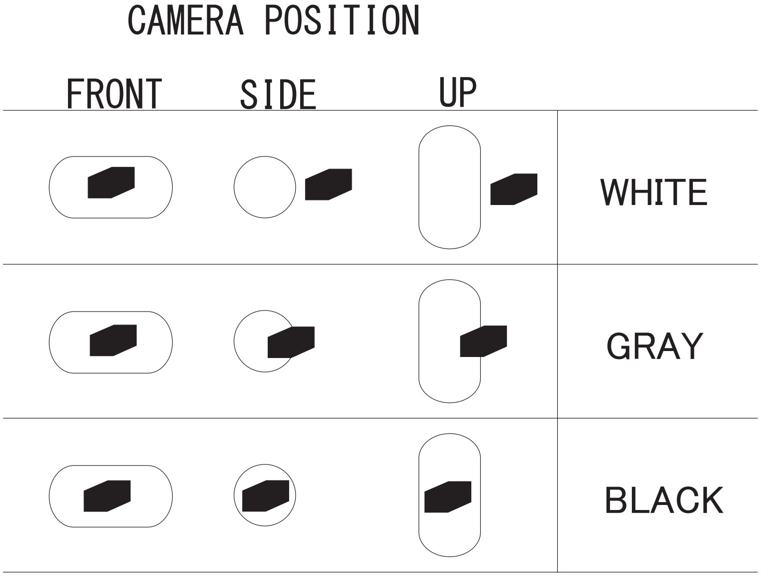


Figure A.3: Classification of octant depending on the overlapping of its projected shadow and the object's silhouette. For clarity, the object's silhouette is rendered as transparent and the hexagon is rendered as black.

B. Link and Surface Movement

The hand model is composed of two main parts: the surface structure and the link structure. The motion of the link structure is always accompanied by a change in the surface structure. Conversely, a change in the hand shape warrants a change in the underlying link structure.

Figure B.1 illustrates how the two structures move together. Each section of the surface structure or voxel data is first assigned to a link. When a link moves, the surface structure (voxel data) also moves to maintain its original configuration with respect to the link.

However, bending a link would cause a tear in the surface structure at the joint as shown in Figure B.2. The figure shows four camera viewpoints of the index finger with its CM, MCP, PIP, and DIP joints bent at certain angles. The crack at the joints are indicated by the dashed circles. Fortunately, for both the model-fitting and the UKF, pose estimation is not affected by the presence of the crack in the surface structure. In the model fitting step described in Section 3, the virtual force to be applied to a link is computed using the voxels along the length

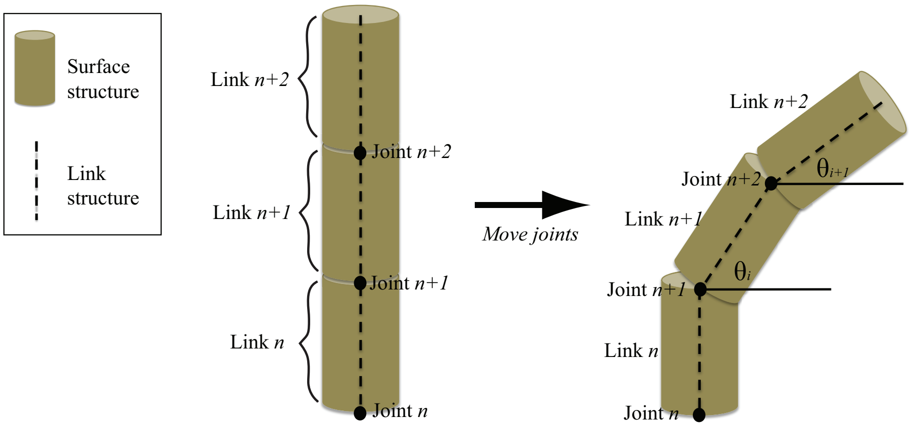


Figure B.1: Illustration of how the link and surface structure move together. Different parts of the surface structure (voxel data) are assigned to the links. When a link moves, the associated surface structure moves along with it.

of the link. In the error measurement step in UKF in Section 4, the distance is measured between the quadrics wrapped around the link and the nearest voxel.

C. Hand Data for Select Users

For the dataglove measurement comparison in Chapter 3, three users were used in the experiments. The images of all the user's hands are included here for visual comparison (see Figure C.1). The link lengths of each user's hand are measured manually and presented in Table C.1.

D. Unscented Kalman Filter

Time Update For a given tracking problem, consider the state dynamics,

$$\mathbf{X_k} = f(\mathbf{X_{k-1}}, \mathbf{R_k}) \tag{D.1}$$

where:
f is the system dynamic,
$\mathbf{X_k}$ is the state vector of size n at time k, and
$\mathbf{R_k}$ is the state noise covariance.

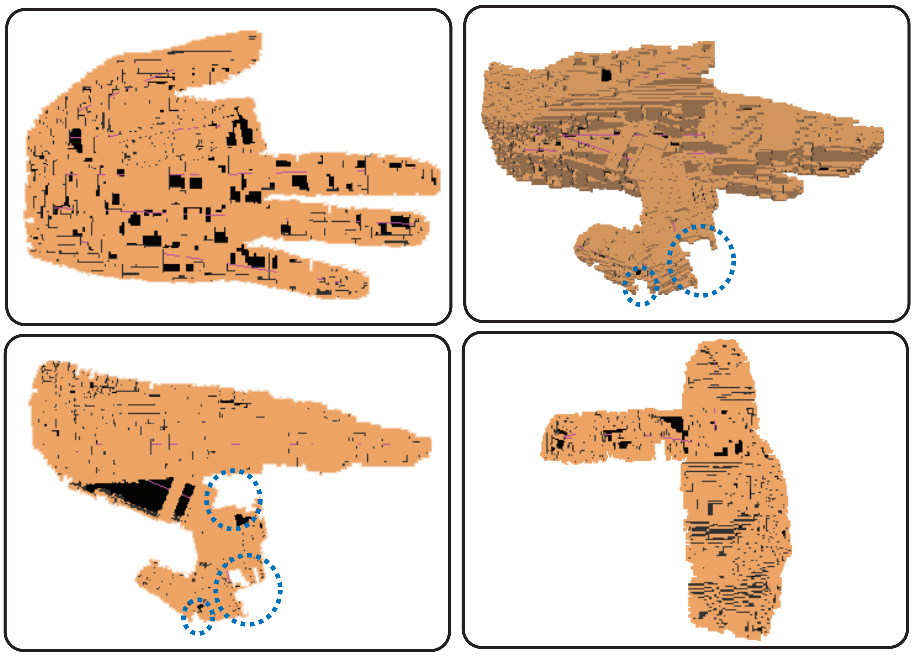


Figure B.2: A bent index finger shown in four different camera viewpoints. The bent joints are CM, MCP, PIP and DIP. Since voxels are associated to specific links, cracks at the joint area develop as the link moves. The dashed circles pinpoint the location of the crack of the joints.

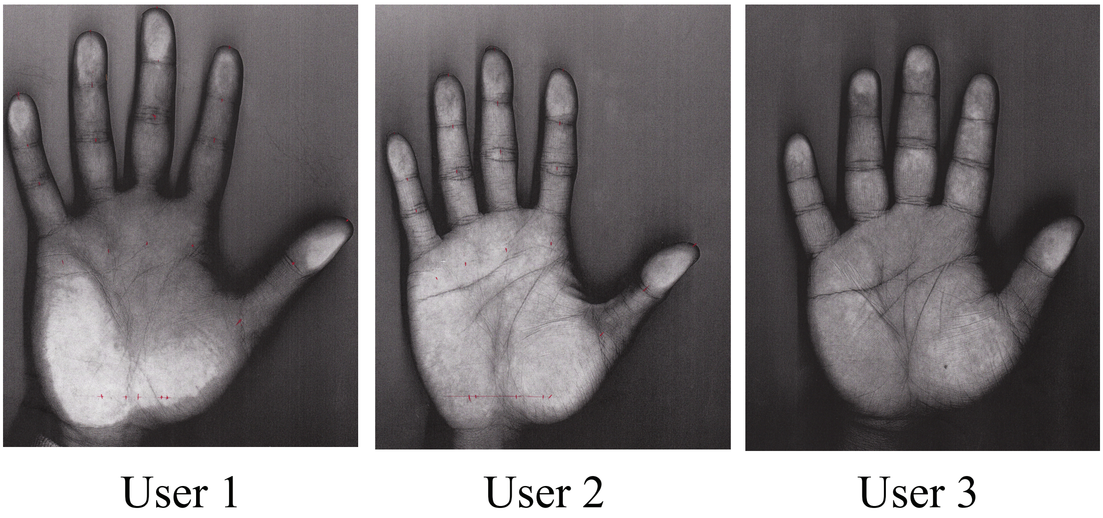


Figure C.1: Hands of three different user scanned. The hands have different physical characteristics; the finger link lengths, the finger girth, and the over-all hand size differ from user to user.

Table C.1: Hand data for some users. Measurements are in millimeters.

Finger	CM-MCP	MCP-PIP	PIP-DIP	DIP-Tip
User 1				
Index	74.0	51.5	20.0	25.5
Middle	74.0	61.0	27.0	26.5
Ring	70.5	53.5	26.0	26.0
Pinkie	67.5	39.5	20.0	25.0
User 2				
Index	75.0	36.5	21.5	25.0
Middle	70.5	47.5	23.5	26.5
Ring	63.0	44.5	22.0	25.0
Pinkie	58.5	33.5	16.0	23.5
User 3				
Index	74.5	37.0	20.5	22.5
Middle	72.5	44.5	25.0	23.5
Ring	68.0	39.0	23.0	24.0
Pinkie	57.5	32.5	15.5	22.5

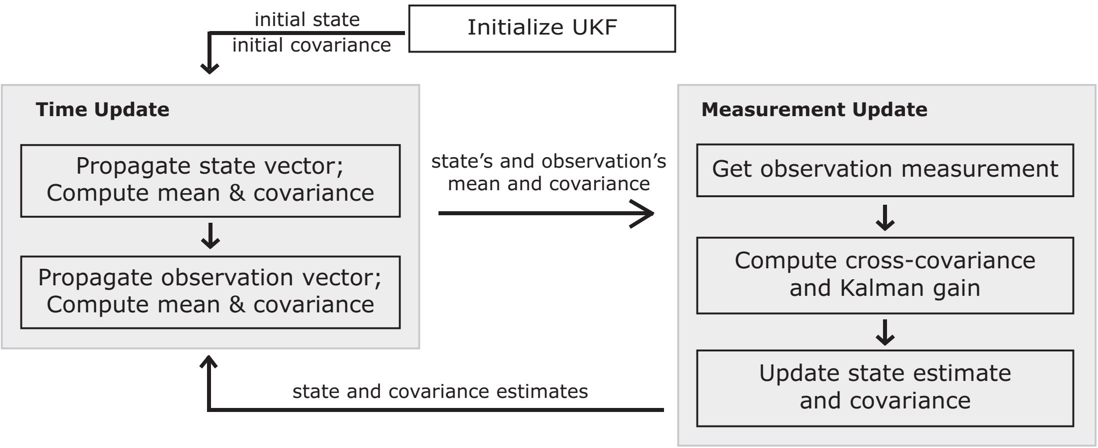

Initialize UKF
initial state
initial covariance
Time Update
Propagate state vector; Compute mean & covariance
Propagate observation vector; Compute mean & covariance
state's and observation's mean and covariance
Measurement Update
Get observation measurement
Compute cross-covariance and Kalman gain
Update state estimate and covariance
state and covariance estimates

Figure D.1: Summary of the Unscented Kalman Filter (UKF) process.

UKF makes an initial estimate of the state vector by selecting sigma points through Equation 4.2:

$$\mathbf{X_k^i} = \begin{cases} \mathbf{X_k^0} = \mathbf{\bar{X}_{k-1}} \\ \mathbf{X_k^i} = \mathbf{\bar{X}_{k-1}} - (\phi)_i & i = 1, \ldots, n \\ \mathbf{X_k^i} = \mathbf{\bar{X}_{k-1}} + (\phi)_{i-n} & i = n+1, \ldots, 2n \end{cases} \tag{D.2}$$

where:
ϕ is the i_{th} column of $\sqrt{(n+\lambda)\mathbf{P_{k-1}}}$,
$\mathbf{P_{k-1}}$ is the covariance estimate from the previous iteration,
$\mathbf{\bar{X}_{k-1}}$ is the state estimate from the previous iteration, and
λ is the scaling parameter.

$2n+1$ sigma points are selected to approximate the posterior mean and covariance of the state vector. The selection of the sigma points is deterministic and is set by adjusting the scaling parameter λ:

$$\lambda = \alpha^2(n+\kappa) - n \tag{D.3}$$

where:
α determines the distribution of the points around the mean and is set to a small positive value, and
κ is a secondary parameter set to 0 or $3-n$.

Equation 4.1 is applied to $\mathbf{X_k^i}$ to obtain the propagated state vector $\mathbf{\hat{X}_k^i}$:

$$\mathbf{\hat{X}_k^i} = f(\mathbf{X_k^i}, \mathbf{R_k^i}) \tag{D.4}$$

The mean $\bar{\hat{\mathbf{X}}}_{\mathbf{k}}$ and the covariance $\hat{\mathbf{P}}_{\mathbf{k}}$ of the propagated sigma points are computed:

$$\bar{\hat{\mathbf{X}}}_{\mathbf{k}} = \sum_{i=0}^{2n} W_i \hat{\mathbf{X}}^{\mathbf{i}}_{\mathbf{k}} \tag{D.5}$$

$$\hat{\mathbf{P}}_{\mathbf{k}} = \sum_{i=0}^{2n} W_i \left[\hat{\mathbf{X}}^{\mathbf{i}}_{\mathbf{k}} - \bar{\hat{\mathbf{X}}}_{\mathbf{k}}\right] \left[\hat{\mathbf{X}}^{\mathbf{i}}_{\mathbf{k}} - \bar{\hat{\mathbf{X}}}_{\mathbf{k}}\right]^T \tag{D.6}$$

The weight W_i is computed according to the following:

$$\begin{array}{ll} W_0 = \{\lambda/(n+\lambda)\} + (1-\alpha^2+\beta) & \\ W_i = 1/\{2(n+\lambda)\} & i = 1, 2, ..., 2n. \end{array} \tag{D.7}$$

β is used to include information about the distribution of the state variable. It is found to be optimal at $\beta = 2$ for a Gaussian distribution.

Likewise, the observation vector is propagated using the propagated sigma points:

$$\hat{\mathbf{Y}}_{\mathbf{k}} = h(\hat{\mathbf{X}}_{\mathbf{k}}, \mathbf{S}_{\mathbf{k}}) \longrightarrow \hat{\mathbf{Y}}_{\mathbf{k}} \approx \left\{\hat{\mathbf{Y}}^{\mathbf{0}}_{\mathbf{k}}, \hat{\mathbf{Y}}^{\mathbf{1}}_{\mathbf{k}}, \hat{\mathbf{Y}}^{\mathbf{2}}_{\mathbf{k}}, \ldots, \hat{\mathbf{Y}}^{\mathbf{2N}}_{\mathbf{k}}\right\} \tag{D.8}$$

where:
h describes the nonlinear observation function,
$\hat{\mathbf{Y}}_{\mathbf{k}}$ is the propagated observation vector,
$\hat{\mathbf{X}}_{\mathbf{k}}$ is the propagated state vector, and
$\mathbf{S}_{\mathbf{k}}$ is the measurement noise covariance.

Then $\bar{\hat{\mathbf{Y}}}_{\mathbf{k}}$, the mean of the propagated observation vector, and its covariance $\hat{\mathbf{P}}_{\mathbf{yy_k}}$ are calculated using the same weights defined in Equation D.7:

$$\bar{\hat{\mathbf{Y}}}_{\mathbf{k}} = \sum_{i=0}^{2n} W_i \hat{\mathbf{Y}}^{\mathbf{i}}_{\mathbf{k}} \tag{D.9}$$

$$\hat{\mathbf{P}}_{\mathbf{yy_k}} = \sum_{i=0}^{2n} W_i \left[\hat{\mathbf{Y}}^{\mathbf{i}}_{\mathbf{k}} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}}\right] \left[\hat{\mathbf{Y}}^{\mathbf{i}}_{\mathbf{k}} - \bar{\hat{\mathbf{Y}}}_{\mathbf{k}}\right]^T \tag{D.10}$$

Measurement Update During the measurement update part, the observation vector $\mathbf{Y_k}$ is obtained from sensor measurements. Then the cross-covariance of the state and the observation vectors, $\mathbf{\hat{P}_{xy_k}}$, is calculated in order to derive the Kalman gain $\mathbf{K_k}$.

$$\mathbf{P_{xy_k}} = \sum_{i=0}^{2n} W_i \left[\mathbf{\hat{X}_k^i} - \mathbf{\bar{\hat{X}}_k}\right]\left[\mathbf{\hat{Y}_k^i} - \mathbf{\bar{\hat{Y}}_k}\right]^T \tag{D.11}$$

$$\mathbf{K_k} = \mathbf{P_{xy_k}}\mathbf{\hat{P}_{yy_k}^{-1}} \tag{D.12}$$

Finally, the state and covariance estimates are updated:

$$\mathbf{\bar{X}_k} = \mathbf{\bar{\hat{X}}_k} + \mathbf{K_k}(\mathbf{Y_k} - \mathbf{\bar{\hat{Y}}_k}) \tag{D.13}$$

$$\mathbf{P_k} = \mathbf{\hat{P}_k} - \mathbf{K_k}\mathbf{\hat{P}_{yy_k}}\mathbf{K_k}^T \tag{D.14}$$

where $\mathbf{\bar{X}_k}$ is the state estimate, and $\mathbf{P_k}$ is its covariance at time k. These values become the input to the next iteration, i.e., $\mathbf{\bar{X}_k}$ becomes $\mathbf{\bar{X}_{k-1}}$ and $\mathbf{P_k}$ becomes $\mathbf{P_{k-1}}$. Then the whole process repeats again.

Upon initialization of the filter, $\mathbf{\bar{X}_{k-1}}$ and $\mathbf{P_{k-1}}$ in Equations 4.1 and 4.2 are set to some initial values and become $\mathbf{\bar{X}_0}$ and $\mathbf{P_0}$, respectively. The scaling parameter values are adjusted heuristically.

For further discussion and details on the implementation of UKF, consult Julier & Uhlmann [52] and Wan & Van der Merwe [59].

E. Camera Set-up for Simulation Experiment in Chapter 4

Figure E.1 illustrates the camera position used in the experiments in Section 4. A total of eight cameras were positioned around the hand to minimize occlusion of the hand. In the experiments involving uncalibrated model, up to eight cameras were used.

For the calibrated model, only four cameras were used: Camera 0, 1, and 2. The fourth camera is positioned at the same level as Camera 3, but on the opposite side such that it sees the hand from the wrist.

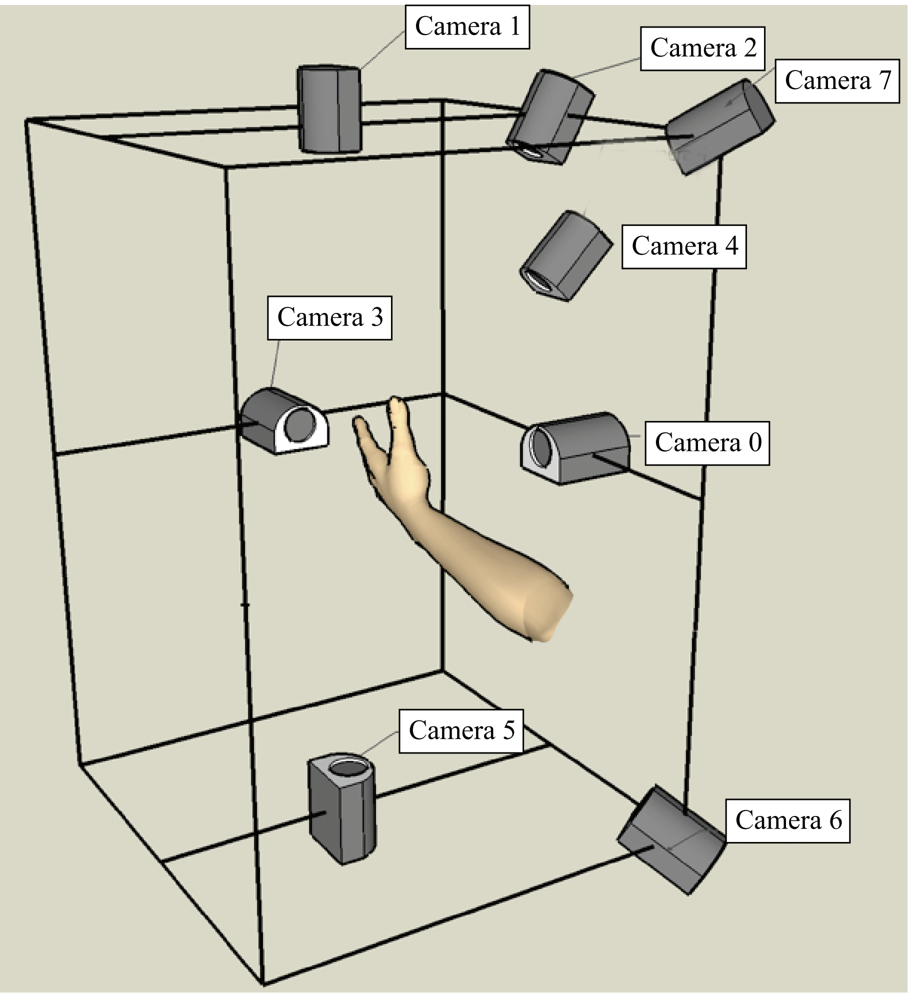


Figure E.1: Camera set-up for the experiments in Section 4.