3D Hand Tracking

RUDRA P K POUDEL

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of

Doctor of Philosophy

August, 2014

Abstract

3D Hand Tracking

The hand is often considered as one of the most natural and intuitive interaction modalities for human-to-human interaction. In human-computer interaction (HCI), proper 3D hand tracking is the first step in developing a more intuitive HCI system which can be used in applications such as gesture recognition, virtual object manipulation and gaming. However, accurate 3D hand tracking, remains a challenging problem due to the hand's deformation, appearance similarity, high inter-finger occlusion and complex articulated motion. Further, 3D hand tracking is also interesting from a theoretical point of view as it deals with three major areas of computer vision- segmentation (of hand), detection (of hand parts), and tracking (of hand). This thesis proposes a region-based skin color detection technique, a model-based and an appearance-based 3D hand tracking techniques to bring the human-computer interaction applications one step closer. All techniques are briefly described below.

Skin color provides a powerful cue for complex computer vision applications. Although skin color detection has been an active research area for decades, the mainstream technology is based on individual pixels. This thesis presents a new region-based technique for skin color detection which outperforms the current state-of-the-art *pixel-based* skin color detection technique on the popular *Compaq dataset* (Jones & Rehg 2002). The proposed technique achieves 91.17% true positive rate with 13.12% false negative rate on the *Compaq dataset* tested over approximately 14,000 web images.

Hand tracking is not a trivial task as it requires tracking of 27 degrees-of-freedom of hand. Hand deformation, self occlusion, appearance similarity and irregular motion are major problems that make 3D hand tracking a very challenging task. This thesis proposes a model-based 3D hand tracking technique, which is improved by using proposed depth-foreground-background feature, palm deformation module and context cue. However, the major problem of model-based techniques is, they are computationally expensive. This can be overcome by discriminative techniques as described below.

Discriminative techniques (for example *random forest*) are good for hand part detection, however they fail due to sensor noise and high inter-finger occlusion. Additionally, these techniques have difficulties in modelling kinematic or temporal constraints. Although model-based descriptive (for example *Markov Random Field*) or generative (for example *Hidden Markov Model*) techniques utilize kinematic and temporal constraints well, they are computationally expensive and hardly recover from tracking failure. This thesis presents a unified framework for 3D hand tracking, using the best of both methodologies, which out performs the current state-of-the-art 3D hand tracking techniques.

The proposed 3D hand tracking techniques in this thesis can be used to extract accurate hand movement features and enable complex human machine interaction such as gaming and virtual object manipulation.

Contents

List of Figures

List of Tables

Acknowledgments

First to many people who helped me and contributed to give the thesis present shape. I owe them all a debt of gratitude. All listed below deserve special mention and thanks.

My sincere thanks to Dr. Hammadi Nait-Charif and Prof. Jian J Zhang, my supervisors, who supervised my work from the very beginning from collection of vague ideas to the completion of this thesis as it stands now; for their guidance, advice, and criticisms during this work and moreover, their encouragement when I felt low.

I am sure without my mother Mandhari Poudel and my wife Anita this would be simply impossible. I owe you both a depth of gratitude. And I immensely miss my father Late Khim Lal Poudel.

My thanks goes to my colleagues at the University Lab, particularly, Arun, Chung, Denis, Jason, Jose, Kathryn, Kripesh, Mat, Min, Sola, Richard, Tauheed, Wenxi, ... and last but not least research administrator Jan Lewis.

My special thanks goes to Bournemouth University for BU Studentship to support my PhD.

Declaration

This thesis has been created by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated. The materials related to region-based skin color technique have been published in Poudel et al. (2012) and Poudel et al. (2013b). The work regarding to the combination of discriminative and descriptive techniques for 3D hand tracking appeared in Poudel et al. (2013a).

Acronym

2D	2 Dimensions
3D	3 Dimensions
CF	Classification Forest
CMC	Carpometacarpal
CPU	Central Processing Unit
CRF	Conditional Random Field
DOF	Degrees Of Freedom
EKF	Extended Kalman Filter
EM	Expectation Maximization
GPGPU	General-Purpose computing on Graphics Processing Unit
GPU	Graphics Processing Unit
HF	Hough Forest
HCI	Human Computer Interaction
HMM	Hidden Markov Model
HSV	Hue Saturation Value
ICP	Iterative Closest Point
IP	Interphalangeal
IR	Infrared
ISM	Implicit Shape Model
KD-Tree	K-Dimensional Tree
MCP	Metacarpophalangeal
MRF	Markov Random Field
MSE	Mean Square Error
PCA	Principal Component Analysis
RGB	Red Green and Blue Color
RF	Regression Forest
SP	Superpixel

SVD	Singular Value Decomposition
TM	Tapeziometacarpal
UKF	Unscented Kalman Fliter

Dedicated to:

My Parents

who taught me how to speak, how to learn ...

Chapter 1

Introduction

Human beings use different types of gestures apart from the voice for human to human interactions, such as hand gestures, body gestures and facial expressions. Hands are the most natural non-spoken means of communication among humans and a major medium of communication with deaf people. Hand gesture is defined as a purposeful movement of the hand (Hassanpour et al. 2008), which carries a given meaning (Poudel 2009). Even though there has been extensive of progress in *human-computer interaction* research in last two decades, it is still largely dependent upon the mouse and keyboard. Manipulating objects on the computer screen using the hand gestures, as Tom Cruise did in a science fiction movie *Minority Report*, is still a dream yet to realized. Such a futuristic human-machine interaction technique inspires human-computer interaction researchers. The major challenge in hand gesture recognition involves four important problems of computer vision and machine learning- segmentation (eg. hand), detection (eg. hand parts), tracking (eg. hand) and learning motion dynamics (eg. gestures). Hence, the hand gesture recognition problem is one of practical, as well as, theoretical importance.

1.1 Motivation

The hand is often considered as one of the most natural and intuitive interaction modalities for human-to-human interaction (Wang et al. 2007). It is also the most natural interaction interface with the physical world because it is used to manipulate objects through grasping, pushing and twisting (Caridakis et al. 2010). However, *human-machine interaction* is still heavily dependent upon the mouse, the keyboard, remote controls, and touch panels as has been the case from the early days of computer technology. Although there has been substantial progress in *human-computer interaction* and *machine learning*, research in the last two decades, the actual methods of human-machine interaction remain largely unchanged.

The contribution to society of a hand gesture recognition system is the main motivation of this thesis. Two example applications where hand gesture recognition can contribute to society are listed below:

1. **Hand gesture recognition technique in mobile devices:** suppose if there was a hand gesture recognition application on a mobile phone, then a deaf person at one end could communicate with another person at the other end who does not understand sign language. A gesture recognition system would translate the signs into the texts, and nowadays most of mobile phones from Microsoft, Google and Apple already have voice to text and text to voice software.

2. **Hand gesture recognition technique for operating theaters:** during surgical operation doctors can use the voice and hand gesture control techniques to control medical devices to prevent contamination (MediKinect 2013). Similarly, such techniques can be used to handle the equipment in a Radiology department from a distance to prevent radiation exposure (Johnson et al. 2011).

In *human-computer interaction* (HCI), proper *3D hand tracking* is the first step in developing a more intuitive HCI system which can be used in applications such as virtual object manipulation and gaming, for example *HoloDesk* (Hilliges et al. 2012) and *hand movement games* (Games 2013) respectively. In recent years, built-in cameras in most consumer electronic devices and the low price of depth sensors have opened new venues for hand gesture recognition research and applications. 3D hand gesture recognition, which is directly dependent on the accuracy of hand tracking, remains a challenging problem due to the hand's deformation, appearance similarity, high inter-finger occlusion and complex articulated motion. Further, 3D hand tracking is also interesting from a theoretical point of view. This is because it deals with the following major areas in computer vision: *segmentation* (of hand); *detection* (of hand parts); *tracking* (of hand); and *occlusion handling* (of hand parts). Hence, improvement of 3D hand tracking for human-machine interaction is a major motivational factor in this thesis.

In summary, the major motivations of 3D hand tracking research can be name as:

1. 3D hand tracking has huge potential to advance human-machine interaction, and has great commercial and social value.

2. Tracking a high dimensional deformable object is still a challenging problem to resolve.

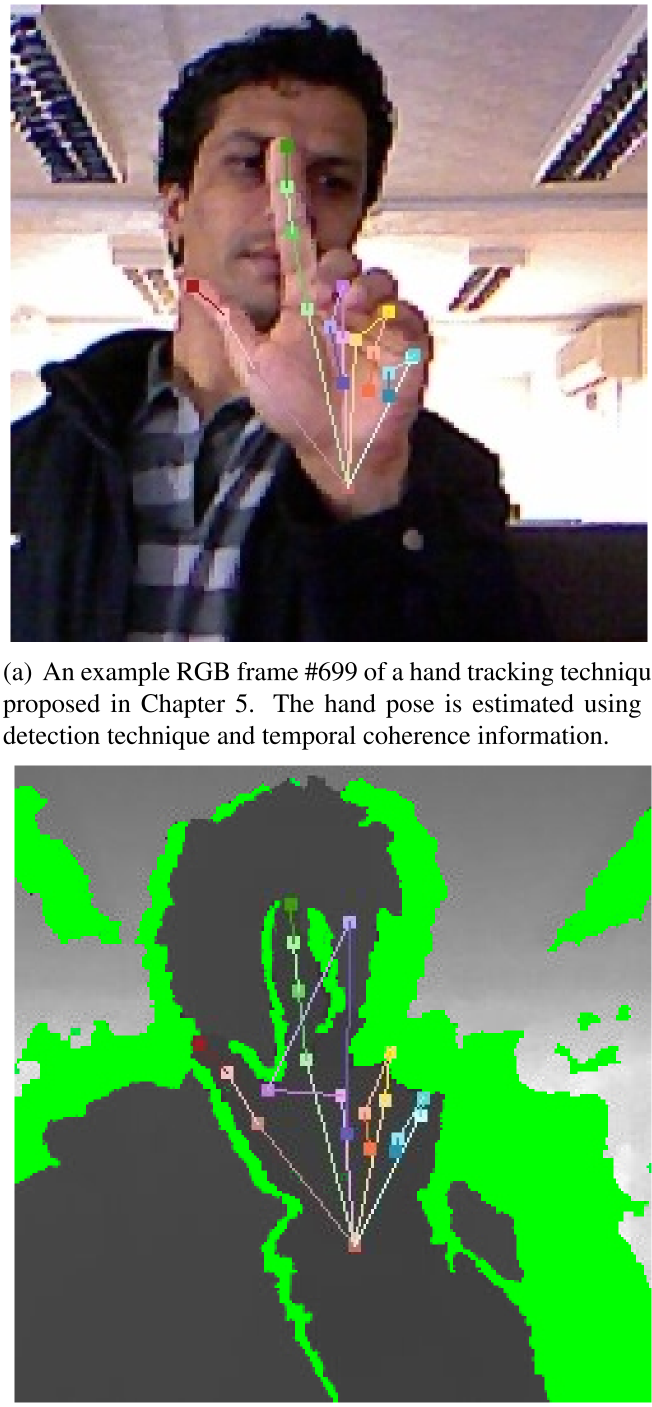
3. Current advancement on depth sensors is opening the door for new possibilities.

1.2 Background

Hand gesture recognition research has received massive attention since the mid nineties, as it will not only revolutionize the human-machine interaction but would also enable communication with deaf people (Rehg & Kanade

1993). Lately, the low cost of the depth sensor and advancement in computing capacity are major motivations for a vision-based hand gesture recognition system without any markers or additional devices. Hand gestures are formed with single pose or multiple hand poses (also know as hand dynamics). Hence, the hand gesture recognition problem may consecutively be divided into two sub-problems namely: i) *Hand Pose Estimation*, and, ii) *Hand Gesture Recognition*. Even though both problems are important, as hand pose estimation is the first-step/precursor to the hand gesture recognition, this thesis consequently focuses on the hand pose estimation and tracking problems.

The availability of depth sensors in recent years has eliminated the difficulties of setting up multiple cameras to acquire the depth for an example Kinect (2013). Also, depth information helps to overcome illumination problems. Hence, in this thesis Kinect (2013) has been used. However, due to reflection, motion and the Kinect sensor's noise itself, depth images used tend to be corrupted (Nguyen et al. 2012), Leading lead to incorrect pose estimation (ref. Figure 1.1(b) on the following page). In such cases, we can improve the accuracy of pose estimation using information from previous poses. The accuracy of hand pose estimation can further improve using the hand kinematic information. Hence, this thesis focuses on 3D hand tracking using hand pose estimation and motion history techniques. In general, the hand tracking techniques exploit the motion coherence information on top of the hand pose estimation techniques.



(a) An example RGB frame #699 of a hand tracking technique proposed in Chapter 5. The hand pose is estimated using a detection technique and temporal coherence information.

(b) An example depth frame #699 of the hand pose estimation technique proposed in Chapter 5. The hand pose is estimated using a detection technique only.

Figure 1.1: *Sub-figures (a) and (b) show the RGB and depth images of frame #699. In Panel (b) depth information of little, index and middle fingers are corrupted. More accurate hand pose estimation is found in Panel (a) than that in panel (b), which demonstrates the advantage of hand tracking technique over hand pose detection technique.*

Hand pose estimation techniques can be divided into two major categories: *appearance-based* and *model-based* techniques (Erol et al. 2007). Appearance-based techniques (Rosales et al. 2001; Athitsos & Sclaroff 2003; Wu et al. 2005; Shotton et al. 2011; Keskin et al. 2011) extract features from an image then map it into a predefined hand pose configuration; hence the quality of the hand pose estimation depends mainly on the robustness of the features. Model-based techniques (Rehg & Kanade 1995; Stenger et al. 2001; Sudderth et al. 2004a; Martin et al. 2008; Hamer et al. 2009; Oikonomidis et al. 2011a) first sample the 3D model of the hand and evaluate it against the observed data. Model-based techniques extract the hand configuration more accurately than appearance-based techniques; however, model-based techniques are computationally expensive. Moreover, based on how individual hand parts are used to estimate the hand pose, hand tracking techniques can be divided into two categories, *joint evidence techniques* where the whole hand is sampled and evaluated as one object, and *disjoint evidence techniques* where all hand parts are sampled and evaluated separately (Oikonomidis et al. 2011a). Joint evidence techniques (Rosales et al. 2001; Stenger et al. 2001; Athitsos & Sclaroff 2003; Martin et al. 2008; Oikonomidis et al. 2011a) efficiently handle the occlusion, but they are computationally very expensive because of the large search space as the hand has 27 degrees-of-freedom. However, disjoint evidence techniques (Rehg & Kanade 1995; Sudderth et al. 2004a; Hamer et al. 2009; Shotton et al. 2011; Keskin et al. 2011) are computationally efficient because they reduce the search space but need additional mechanisms to handle the occlusions and collisions. The unified framework presented in this thesis falls under the appearance-based and disjoint evidence techniques but does not require additional occlusion or collision handling mechanisms unlike other disjoint evidence techniques (Sudderth et al. 2004a; Keskin et al. 2011).

Our proposed framework consist of three modules: i) hand region segmentation: which segments the hand region using skin and depth cues; ii) hand pose estimation: which uses a regression forest to estimate the positions of the hand joints; iii) hand tracking: this uses pose estimation, kinematic prior, and temporal information, to track the hand in 3D. Figure 1.2 shows the overview of the proposed 3D hand tracking system in this thesis.

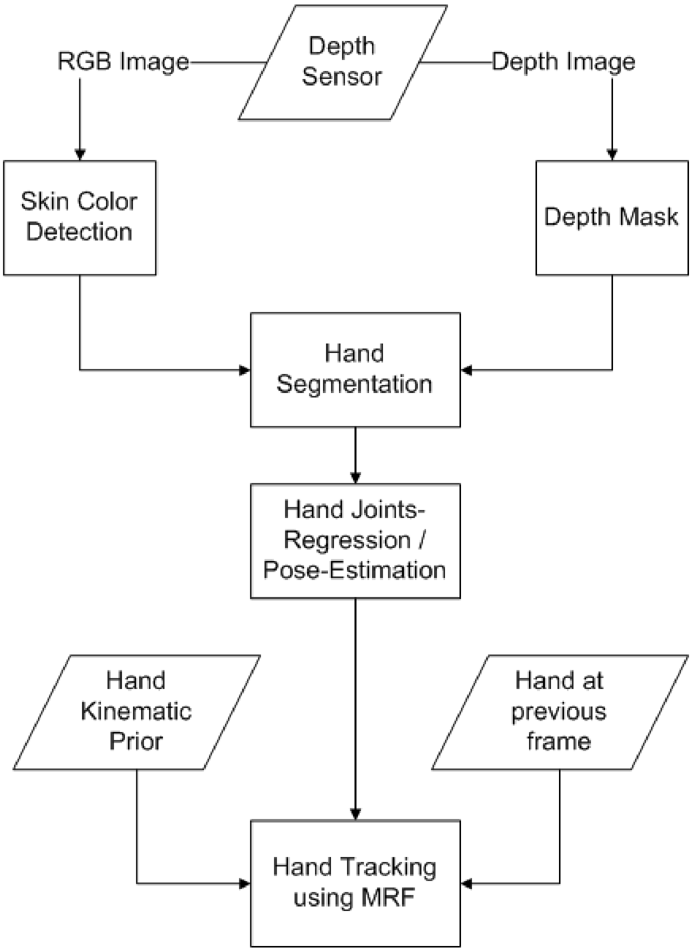


Figure 1.2: *Overview of the proposed 3D hand tracking system.*

Hand Region Segmentation

This thesis uses skin and depth cues for hand region segmentation. Depth helps to overcome the illumination problem and color helps to overcome the depth ambiguity in regard to background objects. Depth cue is provided by the depth sensor, while region-based skin color detection technique has been proposed for the skin cue.

Most of the skin color detection techniques are *pixel-based*, which treat each skin or non-skin pixel individually without considering its neighbours. However, it is natural to treat skin or non-skin as regions instead of individual pixels. Surprisingly, there are only a few region-based skin detection techniques: Yang & Ahuja (1998), Kruppa et al. (2002), Jedynak et al. (2003) and Sebe et al. (2004). Kruppa et al. (2002) and Yang & Ahuja (1998) searched for elliptical skin color shape to find the face. Sebe et al. (2004) used fixed 3x3 pixel patches to train a Bayesian network for skin color detection and Jedynak et al. (2003) smoothed the pixel-based skin color detection results using a *hidden Markov model*. This thesis proposes a new technique exclusively based on the concept of regions, irrespective of the underlying geometrical shape of the target object or predefined rigid shape; for example 3x3 pixel patches. As such, this technique can be easily integrated into any skin detection based system.

The proposed technique uses a segmentation technique called *superpixel* (Moore et al. 2008; Ren & Malik 2003), to group similar color pixels together. This thesis uses ”The Superpixel extraction library” Vedaldi & Fulkerson (2008) for superpixel segmentation. Each superpixel is then classified as skin or non-skin by aggregating pixel-based evidence obtained using a histogram-based Bayesian classifier similar to Jones & Rehg (2002). The result is further improved with *Conditional Random Field* (CRF) (Lafferty et al. 2001), which operates over superpixels instead of pixels. Although the segmentation cost is an additional overhead, that is not involved in the pixel-based approach, it greatly reduces the processing cost further down the line, such as smoothing with CRF. Aggregation of pixels into regions helps to reduce local redundancy and the probability of merging unrelated pixels (Soatto 2009). As superpixels preserve the boundary of the objects, it helps to achieve accurate object segmentations (Fulkerson et al. 2009).

Hand Pose Estimation

The proposed hand pose estimation module uses a discriminative *random forest* (Breiman 2001) to classify the hand-parts and learn joints offsets. Since the voted joint offsets are multimodal in nature, a *mean-shift* (Comaniciu & Meer 2002) voting aggregation technique is used. Unlike Girshick et al. (2011) who selected human body joint proposals independently, this thesis optimizes joint proposals with kinematic prior and temporal constraints globally with a *Markov random Field* (MRF) (Yedidia et al. 2005). We added temporal information on the same semantic level and modelled as MRF.

Hand Tracking

Hand tracking by pose estimation would have been an ideal solution as tracking by detection can overcome the *drifting problem*. Keskin et al. (2011) have recently explored in this direction. However, due to depth sensor noise (Nguyen et al. 2012), some parts of the data are corrupted, further adding to the hand pose estimation error shown in Figure 1.1(b) on page 5. The use of kinematic prior and motion history information can overcome these problems, shown in Figure 1.1 on page 5. Hence, this thesis proposes a novel way of combining hand pose estimation, kinematic prior and motion history information (ref. Chapter 5).

1.3 Research Scope

This section describes the scope of the proposed hand tracking techniques. The aim of this thesis is to track a single hand in an unconstrainted environment. The proposed 3D hand tracking techniques use a Kinect (2013) sensor. However, any depth sensor which produces *RGB* and *depth* images can be used instead. The right hand has been chosen to demonstrate the hand tracking but the proposed techniques are equally applicable for left hand tracking too. Moreover, this thesis does not consider hand manipulating objects, as in Hamer et al. (2009) and Ballan et al. (2012), nor the recent interest of two hand tracking (e.g. Oikonomidis et al. (2012)).

1.4 Contributions

The major contributions of this thesis are:

1. A region-based skin color detection technique, which outperforms the current state-of-the-art pixel-based technique (Poudel et al. 2012).
2. A unified framework for 3D hand tracking which combines discriminative (for example *random forest*) and descriptive (for example *Markov random field*) techniques.

Other contributions are:

1. The use of unexplained observation (segmented hand region minus region covered by predicted hand model) to increase the accuracy of hand joints predictions.
2. Palm deformation module- a module to handle the variations in shape and size of the hand while changing from open to closed shape and vice-versa.
3. Comparison of the classification forest and regression forest techniques for 3D hand tracking.
4. Comparative study of model-based and appearance-based techniques.

The details of the major contributions are explained below.

1.4.1 Skin is better represented as region

Skin color provides a powerful cue for complex computer vision applications. Although skin color detection has been an active research area for decades, the mainstream technology is based on individual pixels. This thesis presents a new region-based technique for skin color detection which outperforms the current state-of-the-art *pixel-based* skin color detection technique on the popular *Compaq dataset* (Jones & Rehg 2002). A color and spatial distance based clustering technique is used to extract the regions from the images, known as *superpixels*. In the first step, the proposed technique uses the state-of-the-art non-parametric pixel-based skin color classifier (Jones & Rehg 2002), which we call basic skin color classifier. The pixel-based skin color evidence is then aggregated to classify the superpixels i.e. regions. Finally, a *Conditional Random Field* (CRF) is applied to further improve the classification result. As CRF operates over superpixels, the computational overhead is minimal. However, any good pixel-based or region-based skin color method can be used as a basic skin color classifier.

1.4.2 Combining discriminative and descriptive techniques

Discriminative techniques are good for hand part detection but they fail due to noisy data (Nguyen et al. 2012) and high inter-finger occlusion. In addition, these techniques do not incorporate any kinematic or temporal constraints. Even though model-based descriptive (for example *Markov Random Field*) or generative (for example *Hidden Markov Model*) techniques use kinematic and temporal constraints well, they are computationally expensive, and hardly recover where tracking failures occur. This thesis presents a unified framework for 3D hand tracking, using the best of both methodologies. Hand joints are detected using a *regression forest*, which uses an efficient voting technique for joint location prediction. The voting distributions are multi-modal in nature. Hence, rather than using the highest scoring mode of the voting distribution for each joint separately, we fit the five highest scoring modes of each joint on a tree-structure *Markovian model* along with kinematic prior and temporal information. Experimentally, it has been observed that trusting the discriminative technique (i.e. joints detection), more than descriptive or generative techniques (i.e. history information), produces better results. Therefore, the proposed technique efficiently incorporates this observation by fixing/freezing the 50% high scoring joint positions and searching for the remaining 50% low scoring joint positions, using history and hand kinematic information. This strategy reduces the computational cost and produces good results for 3D hand tracking on RGB-D data.

1.5 Publications Related with This Thesis

The publications related with this thesis are listed below,

1. Poudel R. P. K., Nait-Charif H., Zhang J. J. and Liu D., 2013. Skin Color Detection Using Region-Based Approach. In: *International Journal of Image Processing*, vol. (7), issue (4), pp. 385-394.
2. Poudel R. P. K., Fonseca J. A., Zhang J. J. and Nait-Charif H., 2013. A unified framework for 3D hand tracking. In: *9th International Symposium on Visual Computing*, Crete, Greece, pp 129-139.
3. Poudel R. P. K., Nait-Charif H., Zhang J. J. and Liu D., 2012. Region-based skin color detection. In: *8th International Conference on Computer Vision Theory and Applications*, Rome, Italy, pp. 301-306.

1.6 Thesis Outline

The outlines of the remaining chapters of this thesis are listed below,

Chapter 2 presents a literature overview of hand tracking. 2D and 3D hand tracking research are described separately. The similarities and differences between hand and whole body tracking are also explained. The literature related to skin-color detection techniques is reviewed and the other major algorithms related with this thesis are also explained briefly in this chapter.

Chapter 3 argues that skin is better represented as regions rather than *pixels* and proposes a *region-based skin color detection* technique. A comparison of the proposed region-based technique with a state-of-the-art pixel-based technique is also presented.

Chapter 4 presents a *model-based* 3D hand tracking technique using *Markov random fields*. It also proposes a palm deformation module since the shape of the palm deforms significantly. In addition, to improve the accuracy of the hand tracking, multiple cues have been applied, such as depth difference between segmented hand region and predicted hand model.

Chapter 5 presents an appearance-based 3D hand tracking technique which efficiently combines the discriminative technique (*random forest* (Breiman 2001)) and descriptive technique (*Markov Random Field*). Regression forest has been used to predict the hand joint positions. The prediction of the hand joint positions are further improved using temporal motion coherence and the kinematic information of the hand.

Chapter 6 summaries the major contributions of this thesis and directions for future research.

Chapter 2

Literature Review

Hand tracking related research reviews have been published in Pavlovic et al. (1997), Wu & Huang (1999) and Erol et al. (2007). As hand tracking is an important part of the hand gesture recognition system, hand tracking research reviews can also be found in hand gesture recognition literature (Konstantinos G. 2004; Hassanpour et al. 2008). The research related to object tracking is relevant to the 3D hand tracking research in general, so the next section will overview the visual tracking techniques in general, the hand model used in 3D hand tracking and different types of hand tracking techniques separately. Section 2.2 discusses the similarities and differences between whole human body and the hand tracking techniques. Skin color is an important cue for hand region segmentation and is used in all proposed techniques of this thesis. An overview of the skin color detection techniques is provided in Section 2.3. Section 2.4 describes the techniques specific to this thesis. The final Section 2.5 summaries the whole chapter.

2.1 Hand Tracking

Hand tracking techniques share the common literature with the tracking techniques in general. As the thesis focuses on the vision based hand tracking, this chapter will look at the history of the visual tracking and hand tracking before moving on the vision based hand tracking techniques.

2.1.1 Vision-Based Tracking Techniques

Visual tracking is about localization of a particular object in successive frames of a video sequence. Unlike object localization in each image independently, tracking exploits object dynamics for efficiency and effectiveness using the information from previous frames. Normally, tracking is an online process and therefore the emphasis is on real-time algorithms (Blake 2006). Recent progress in computing and especially fast development of *general-purpose graphics processing unit* (GPGPU) (CUDA 2013; AMD-GPU 2013; OpenCL 2013) have enabled complex algorithms to run in real-time.

Tracker initialization is the first step in object tracking. In this step, object position and tracking parameters are initialized. The initialization can be automatic or manual. At the arrival of each successive frame, tracking then follows the following steps,

1. Based on the object position and dynamics at time $t - 1$, estimate the object position and dynamics at time t.

2. Search for the target object locally (Blake 2006).

If the initialization or parameter estimation in the first step is not good enough, tracking is likely to perform poorly and in the long run it can cause tracking failure known as the *drifting* problem in the tracking literature. Automatic recovery of tracking is a difficult task: normally some sort of reinitialization is required. Tracking by detection or pose-estimation (Shotton et al.

2011; Keskin et al. 2011) is a favoured technique to overcome the drifting phenomenon.

Tracking techniques involve iterative *target matching*, also know as *template matching* (Lucas & Kanade 1981). *Blob tracking* has also received attention in early 2000 due to the computational advantage over template matching technique. *Mean shift* tracking (Comaniciu et al. 2000) is one of the most influential blob based tracking technique. Mean shift algorithm is efficient in processing, but it gives local maxima rather than global maxima (Comaniciu et al. 2000). Parametrized curve matching techniques called *active contours* are other popular techniques for tracking objects. The active contour model called *snake* (Kass et al. 1987) is a popular technique in this category. However, contour is influenced by illumination, so the quality of tracking by active contour is as well (Blake 2006).

Another category of tracking techniques is *filtering* techniques for example *Kalman filter* (Julier & Uhlmann 1997, 2005) and *particle filter* (Isard & Blake 1998a,b,c). Particle filter is robust but computationally expensive as it needed many samples to track a target object accurately. *Extended Kalman Fliter* (EKF) (Julier & Uhlmann 1997) could model non-linear systems more efficiently than particle filter but the inter-state transition is assumed to be linear.*Unscented Kalman Filter* (UKF) (Julier & Uhlmann 2005) improves the shortcomings of EKF by propagating the *mean* and *covariance* using a *Monte Carlo* sampling technique. UKF uses few sampling points (for example 15 to 30) for approximation, weighted covariances are added on half of the samples and subtracted from the remaining half of the samples from the projected sample point i.e. the predicted center of the target object. The approximation obtained from UKF is accurate to the 2nd order of non-Gaussian and 3rd order of Gaussian inputs only.

Features (for example *optical flow* (Barron et al. 1994), *scale invariant feature transform* (SIFT) (Lowe 1991), corner detector (Harris & Stephens

1988), *self-similarity* (Shechtman & Irani 2007) and *histograms of oriented gradient* (HOG) (Dalal & Triggs 2005)) play an important role in tracking. Lucas & Kanade (1981) proposed one of the early tracking technique: using optical flow feature, they matched optical flow features locally for correspondence finding in pairs of images. Lucas & Kanade (1981) used spatial intensity gradient of images and iteratively matched within neighbourhood only for image registration. Zhou et al. (2009) used SIFT features to match the region of interest across consecutive frames and further improved using mean-shift via color histograms. Local image features provide useful information about the temporal information in an image. Laptev (2005) added the local features in space-time to detect walking people. Lai et al. (2010) improved the results further by combining the work of Efros et al. (2003) and Lai et al. (2010). Lai et al. (2010) proposed a displacement feature based on the SIFT. Usually, SIFT features are calculated on all interest points of the images, which hits the performance hard. The *speeded up robust feature* (SURF) (Bay et al. 2006) feature provides an alternative to SIFT. Ta et al. (2009) used the SURF feature for tracking and continuous object recognition.

2.1.2 History of Hand Tracking

In the early days, *mechanical gloves* (Zimmerman et al. 1987; Fels & Hinton 1997) were the only effective tools for capturing hand motions (Sturman & Zeltzer 1994). The gloves are worn on the hand to measure the hand joints positions and movements in real-time. Dorner (1994) and Wang & Popovi (2009) used multi-color marker gloves to track the hand. The major drawback of the glove based technique is the need to wear the glove, which is cumbersome. To overcome the limitation of the glove based hand tracking techniques, researchers have been working on markerless *computer vision* techniques (Rehg & Kanade 1994; Cipolla & Hollinghurst 1996; Stenger et al. 2001; Wu et al. 2005; Hamer et al. 2009; Keskin et al. 2011; Oikonomidis et al. 2010; Poudel et al. 2013a) for hand tracking.

The angle of view affects the shape of the object. To overcome this problem, Starner et al. (1997) and Starner et al. (1998) mounted the camera on the head so that the distance of the hand from camera and angle of view remains more or less constant. They built a wearable cap with a camera mounted on it which pointed towards the ground. Such a technique is also know as *wearable* device/computing. Starner et al. (1998) experimented using a 40 word lexicon. They reported the 92% word accuracy with the camera mounted on a desk and 98% word accuracy with the camera mounted on the cap of the user.

Stenger et al. (2001) used an unscented Kalman filter (Julier & Uhlmann 1997) to track a 3D hand. The experiment shows that the unscented Kalman filter is robust in modelling Gaussian based motion but cannot model the non-Gaussian motion i.e. change of random direction frequently. Stenger et al. (2006) reformulated the 3D hand tracking using hierarchical template matching from a *database*. The problem with this type of technique is that it needs to store all possible templates, which can be expensive. Oikonomidis et al. (2011a) also used a template matching technique but generated the templates online and optimized with a *particle swarm* (Eberhart & Kennedy 1995; Kennedy et al. 2001) technique. Template matching techniques recover the hand configuration well and effortlessly handle the occlusion. However, template matching techniques are computationally expensive as the hand has 27 degrees-of-freedom (DOF) (ref. Section 2.1.4). To narrow down the search space, Sudderth et al. (2004b) matched each hand-part template separately and used *non-parametric belief propagation* (Sudderth et al. 2003) for the global hand configuration optimization. Sudderth et al. (2004a) later added an occlusion handling technique. This kind of technique reduces the search spaces but adds the complexity of occlusion handling. The common problem of template matching is computational cost as they need to test many samples.

On the other hand, detection based techniques (Keskin et al. 2011) infer the hand-model from feature matchings. Hence, the quality of detection based techniques is based on the quality of extracted features, which varies based on the sensor noise and environmental factors.

Based on the taxonomy of Erol et al. (2007), hand tracking techniques can be divided into two broad categories: *appearance-based* and *model-based* techniques. Moreover, based on how individual hand parts are used to estimate the hand pose, hand tracking techniques can be divided into two categories, *joint evidence techniques* and *disjoint evidence techniques* (Oikonomidis et al. 2011a). These are discussed separately in detail in Section 2.1.7 and Section 2.1.8.

2.1.3 Mechanical and Color Gloves

Keyboard, mouse and joystick are the major medium of inputs to the computer applications. The naturalness of how the hand manipulates objects in real life cannot be replicated using such devices. To bring the naturalness in human-computer interaction the *Put-that-there* (Bolt 1980) project started at *Massachusetts Institute of Technology* (MIT) in early 1980's. Bolt (1980) used the commercial *polhemus sensor* Zimmerman et al. (1987) to track the position and orientation of the hand. The *Polhemus sensor* can provide the six degrees-of-freedom by radiating a pulsed magnetic field from a fixed source. Bolt (1980) used hand position and orientation information to select the graphical elements from the screen.

One of the early *data glove* developed by Zimmerman et al. (1987) was able to capture 10 finger joints and six degrees-of-freedom of the hand position and orientation. One of the major advantage of the data glove over camera based systems is that the accuracy of the data glove technique is not affected by the line-of-sight. The data glove technique can capture the rapid motion of the hand and run in real-time, which was the major attraction in the late eighties (Zimmerman et al. 1987).

Kramer & Leifer (1988) developed a thin fabric glove called *CyberGlove*. The small electronics boxes attached to the glove are capable of recoding the positions and sending the digital streams to the computers via standard serial port. They used such digital signals to translate American sign language into spoken English language. They are more stable than the data glove of Zimmerman et al. (1987). Later, many companies commercially produced the CyberGlove, which were capable of handling the complex gestural work.

Recently, wearable devices have once again become popular in commercial industry such as *google glass* (Google 2013) and *Digits* (Kim et al. 2012). The work of Kim et al. (2012), called *Digits*, is one of the latest and impressive examples of wearable devices research. Digits is a wrist-worn sensor, which can fully recover the 3D pose of the user's hand without wearing any gloves. Digits can recover the hand pose while users are moving/walking. Kim et al. (2012) used *infrared* (IR) camera to get the information of the hand shape. They used samples of finger tips and lower regions of the finger and fed into a kinematic model of the hand. The kinematic model applied biomechanical constraints of the hand to recover the accurate 3D pose of the user's hand. Kim et al. (2012) demonstrated the human-computer interaction using a mobile phone and Digits.

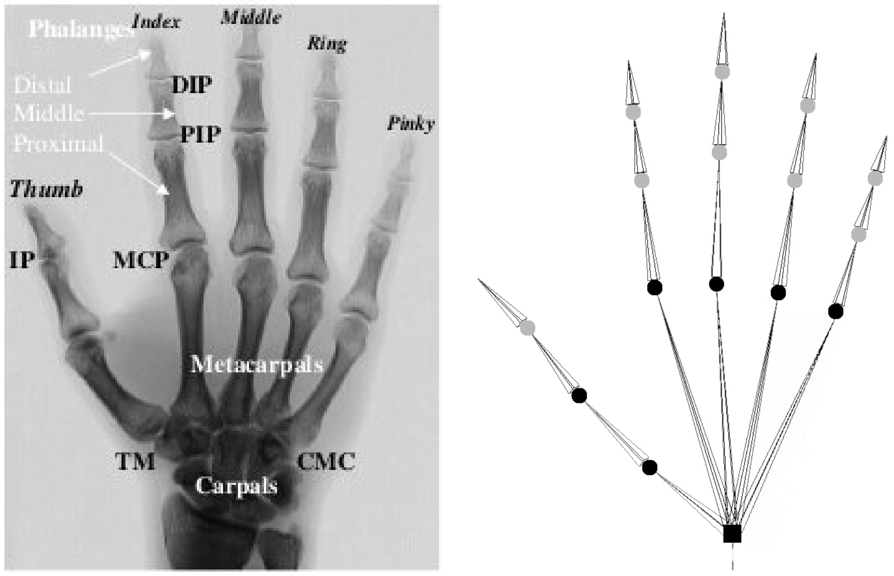
Instead of using electro-magnetic devices or infrared camera, Wang & Popovi (2009) used a multi-colored glove to recover the pose of the hand from a single RGB image. The multi-colored glove was built by using simple ordinary cloth. Each hand part is marked with a different color. Wang & Popovi (2009) used the nearest-neighbour approach to track the hand in real-time. Further, to improve the hand pose accuracy they used inverse kinematics and temporal information. Wang & Popovi (2009) demonstrated sign language alphabet transcription task, virtual object manipulation task and simple character animation systems to prove the importance of their proposed technique.

Mechanical/electronics gloves are reliable and accurate devices for 3D hand tracking. The computer vision based hand tracking technique using a color glove is even more attractive as the color glove is simple and easy to configure. However, these techniques add an extra burden to the user as the glove is needed to be worn. So, making gestures with gloves feels clumsy and does not feel as natural as the naked hand. There are also many situations where wearing a glove is impractical; for example, by doctors in an operating theater to interact with *magnetic resonance imaging* (MRI) or *computed tomography* (CT) systems. Due to the above mentioned problems, computer vision based 3D hand pose estimation and tracking of the naked/markerless hand is an important research problem.

2.1.4 Hand Model

This section provides an overview of the hand kinematic model. The kinematic model represent the motion of the hand skeleton and it is used for 3D hand tracking. Figure 2.1(a) on the following page shows an X-ray of a right hand. The human hand consists of 27 bones. Eights bones are located on the wrist, called carpals. Carpal bones join the fingers with the wrist and are ignored by 3D hand tracking techniques (Wu & Huang 2001; Sudderth et al. 2004b; Hamer et al. 2010). The length of the bone is varied from person to person and degrees-of-freedom depends upon the joint. The names of the joints are based on the connecting bones. The type of joints and their degrees-of-freedom are described below.

- **Carpometacarpal Joints (CMC)**: CMC connects the metacarpals of the fingers with wrist. The CMC of index and middle fingers are static. The ring and pinky/little fingers CMC have limited movement and are



(a) Hand anatomy. The picture is taken from Erol et al. (2007)

(b) Kinematic hand model

Figure 2.1: *Figure (a): anatomy of the right hand. Figure (b): equivalent kinematic model of the right hand. Square box represents 6 DOF of the hand position and orientations. Black circles represent 2 DOF: abduction/spreading. White circles represent 1 DOF: flexion only.*

not considered in the hand tracking problem.

- **Trapeziometacarpal Joints (TM)**: CMC of thumb finger is know as TM. It is difficult to model as it has two non-orthogonal and non-intersecting rotation axes (Hollister et al. 1992; Erol et al. 2007). However, in practice TM is modelled as two degrees-of-freedom.
- **Metacarpophalangeal Joints (MCP)**: MCP connects finger with palm. It has two DOF, one for abduction/spreading-finger and one for flexion.
- **Interphalangeal Joints (IP)**: IP connects finger phalanges/bones. It has one DOF for flexion.

The hand anatomy and equivalent kinematic hand model is shown in Figure 2.1. In model-based hand tracking, 27 DOF are considered: 6 for the hand position and rotation, 2 for a trapeziometacarpal, 10 for five metacarpophalangeal and 9 for nine interphalangeal (ref. Figure 2.1). Such a kinematic model has been used in this thesis.

2.1.5 Appearance-Based Hand Tracking

Appearance-based techniques model a gesture as a sequence of views, and are known as *view-based* techniques. Appearance-based techniques extract features from images then classify them to predefined hand postures (Darrell & Pentland 1993; Cui & Weng 1996; Black & Jepson 1998; Rosales et al. 2001; Gupta et al. 2002; Athitsos & Sclaroff 2003; Wu & Huang 2000; Wu et al. 2005; Zahedi et al. 2005; Yuan et al. 2005; Romero et al. 2009; Keskin et al. 2011; Poudel et al. 2013a). Hence, the quality of the hand tracking mainly depends on the robustness of the features. Appearance based techniques use 2D models and do not extract the exact configuration of the hand. This makes appearance-based techniques more suitable for simple user interfaces, where accurate hand configuration is not necessary. For example, selecting the menu items, sliding the presentation and selecting, playing, and stopping music.

Appearance-based techniques extract a large number of features from the image. Hence, *principal component analysis* (PCA) is the common approach to reduce the features dimension. Black & Jepson (1998) presented a view-based representation for rigid and articulated objects tracking using *eigenspace* and parametrized optical flow estimation. They used an *Eigen-Pyramid* representation and a coarse-to-fine matching technique for large affine transformations between the eigenspace and the image. They were also able to handle occlusion to some extent. They demonstrated hand gestures recognition in video sequences as an example of the proposed technique. The major drawback of the optical flow based technique is that apart from objects of interest, other objects and background are assumed to be more or less static.

MacCormick & Isard (2000) used the *partitioned sampling* (MacCormick

& Blake 2000) technique to track the hand articulation. It is similar to a hierarchical search and avoids the high cost of particle filters when tracking more than one object. MacCormick & Isard (2000) also introduced the *survival rate* module for particle filters (Isard & Blake 1998c) to increase the efficiency of partitioned sampling. MacCormick & Isard (2000) track six DOF of the hand in real-time and had a self-initializing module. They modelled the hand using B-spline, used skin color and contour features for hand tracking. The shortcoming of this technique is that contour and skin color are not reliable features in variable illumination and unconstrainted environments.

Wu & Huang (2000) introduced the *discriminant expectation maximization* (D-EM) algorithm for view independent hand pose recognition. They learned hand poses using supervised and unsupervised learning techniques. As collecting supervised data is difficult, they used the *adaptive self-organizing color segmentation* technique (Wu et al. 2000) to collect large amounts of unlabelled data and manually labelled some hand gesture data. Wu & Huang (2000) used color and edge information after background subtraction to generate *Gabor wavelet filters* and 10 coefficients from the *Fourier descriptor* were used to represent the hand shape. Further Wu & Huang (2000) studied hand pose recognition using D-EM algorithm and compared it with other techniques. Their technique relies on the color and contour features and those features are not robust to the illumination variation.

Zahedi et al. (2005) proposed a hand gesture recognition technique without tracking 3D hand tracking. They used skin color and various variants of derivative between successive frames to generate the hand shape/pose features. They used *hidden Markov model* (HMM) (Rabiner 1990) for gesture learning and experimented with 10 gestures and only had a 7% error rate. The common problem of this type of techniques is that they assume the background is static, which is hardly true is real scenarios.

Wang & Wang (2008) also used a feature based hand pose detection technique for human robot interaction. Wang & Wang (2008) argued that *scale-invariant features transform* (SIFT) (Lowe 1999) features are well suited to represent hand poses in different orientations and learned the hand poses using *boosting* (Freund 1990; Freund & Schapire 1995) technique. However, their technique could not handle the occlusion.

Appearance-based techniques map the hand features to pre-defined hand poses. As these techniques only involve 2D image processing, they are computationally fast and mostly run in real-time. However, these techniques are not robust on hand configuration estimation because they generally cannot deal with occlusion. These techniques are less sensitive to the spatial location of the hand, which is another major drawback of these techniques. Hence, these techniques cannot be used effectively in virtual object manipulation but are more suitable on hand posture recognition.

The following section describes the more complex 3D hand tracking techniques, which can extract the 3D configuration of the hand and can be used for complex tasks such as the virtual object manipulation.

2.1.6 Model-Based 3D Hand Tracking

A 3D hand model mimics the human hand skeleton, which is used to estimate the kinematic parameters of the hand (ref. Figure 2.1 on page 22). Most researchers modelled the 3D hand shape with 27 degrees-of-freedom (DOF). However, some authors modelled the 3D hand with less DOF by imposing further constraints based on the bionic view. DOF can be further restricted based on the relevance of the required gestures for the targeted application. The model-based techniques (Rehg & Kanade 1995; Sudderth et al. 2004a; Martin et al. 2008; Hamer et al. 2009; Oikonomidis et al. 2011b, 2012) first samples the 3D model of the hand and evaluates it against the observed data. This is an inverse matching problem. Hence, searching the optimal values of the hand configuration is computationally expensive and a difficult task. Generally, 3D hand parameters are estimated by edges or depth matching on each frame.

Lowe (1991) proposed the earliest parameterized three-dimensional models. The technique of Lowe (1991) was able to handle objects with arbitrary curved surfaces and any number of internal parameters representing articulations, variable dimensions or surface deformations. Searching the projection of parameterized three-dimensional hand model had been first introduced by Rehg & Kanade (1993). Rehg & Kanade (1993) proposed the earliest model-based 3D hand tracking technique called *DigitEyes*. They used edge and point features to match the projection of the 3D hand model with gray scale video image. DigitEyes used two cameras and tracked the 27-DOF of hand. Further, Rehg & Kanade (1993) demonstrated a simple 3D mouse interaction application using a single camera.

Another early model was proposed by Heap & Hogg (1996): they created a 3D hand using the *point distribution model* (Cootes et al. 1995). They learned the hand deformation and movement using *simplex mesh* (Delingette 1994) and used finger tips as control points to iteratively fit a deformed mesh model in successive frames. Heap & Hogg (1996) achieved real-time tracking for 6 DOF only. Their technique could not handle finger occlusion.

Stenger et al. (2001) presented a more accurate hand modelling technique using ellipsoids, cones and cylinders. Their technique could also deal with occlusion. They matched the edge of the projected 3D hand sample with input video images to infer the hand parameters and used unscented Kalman filter to model the motion dynamics. They were able to achieve 3 frames per second with a single camera on a Celeron 433MHz computer to track the 7 DOF of hand (6 DOF of global hand position and orientation and 1 DOF of thumb). They demonstrated examples of their techniques for multiple cameras.

Template based 3D hand tracking techniques can extract the hand parameters accurately. However, tracking 27 DOF of hand is a computationally expensive task. To increase the efficiency Stenger et al. (2006) proposed the hierarchical template matching technique. They saved all hand templates with known configurations in a database and hierarchically looked up in runtime. Even though the hierarchical template matching technique introduced by Stenger et al. (2006) increased the efficiency, their technique needed large storage capacity and changes in the hand size of the signer causes the reinitialization of the database.

Most of the model-based hand tracking techniques estimate the global hand motion using foreground segmentation or hand movement estimation from the previous frames. They then do an exhaustive search for the local hand parts configurations. Such a strategy requires large number of samples as the hand has 27 DOF. Hence, the computational cost is too high. However, foreground segmentation is not an easy task and hand movement estimation is also difficult as the hand changes the direction too often. To tackle this problem, Wu et al. (2005) applied a *divide-and-conquer* strategy. They learned the hand motion prior for global hand motion estimation and tackled finger articulation using a sequential Monte Carlo tracking algorithm. The sequential Monte Carlo tracking algorithm produced good results but it was still computationally expensive.

Martin et al. (2008) proposed a model based approach to 3D hand tracking using some new features such as example shadow and texture. They estimated the 3D hand configuration from a monocular video. Martin et al. (2008) dynamically estimated the hand texture and the illumination and minimized objective function using a *quasi-Newton* technique. They exploited the texture's temporal continuity and shadow information to improve the hand parameter estimation. They also introduced gradient terms to improve the self-occlusion of the fingers. However, their technique is computationally expensive similar to those of full hand sampling at once (ref. Section 2.1.7) and template matching.

Instead of sampling and evaluating the whole hand at once, Sudderth et al. (2004b) samples and evaluates each hand-part separately. Even though the sum-of DOF of each hand parts is much higher than a hand as a whole, it reduces the search space (Hamer et al. 2009). Sudderth et al. (2004b) used *nonparametric belief propagation* (Isard 2003; Sudderth et al. 2003) to enforce the kinematic constraints of the hand, which they improved later for the occlusion handling technique in Sudderth et al. (2004a). Later Hamer et al. (2009) proposed a hand tracking technique while manipulating an object inspired by the Sudderth et al. (2004b). Hamer et al. (2009) was improved later in Hamer et al. (2010) by including an object-dependent hand pose prior. They learned object-dependent hand pose prior using sparse training data. However, Hamer et al. (2009) used *generalized belief propagation* (Yedidia et al. 2005) instead of nonparametric belief propagation. Basically, the two approaches are very similar: nonparametric belief propagation uses particle filters for sampling with *belief propagation* while in belief propagation the sampling method is independent of the message passing algorithm. Both methods used *local trackers* for 16 hand parts: 3 parts of each of the 5 fingers and one to palm.

In summary, model-based techniques extract the hand configuration more accurately than appearance-based techniques. However, model-based techniques are computationally expensive.

2.1.7 Joint Evidence Techniques

Joint evidence techniques (Wu & Huang 2001; Stenger et al. 2006; Martin et al. 2008; Oikonomidis et al. 2011a) considered the whole hand as a single object. Treating the whole hand as one single object/hypothesis avoids the problem of explicit occlusion handling. Kinematic constraints are inbuilt in the joint evidence techniques. Wu & Huang (2001) presented a *cardboard model* and sampled the whole hand at once. Even though hand has 27 degrees-of-freedom, only certain hand configurations are natural and possible due to kinematic constraints. As such, Wu & Huang (2001) learnt the natural hand articulation to reduce the search space. Also, to deal with the high dimensionality problem, Stenger et al. (2006) stored hand shapes in a database and hierarchically searched at runtime. Storing all possible hand samples in the database increased the cost of storage space.

The recent work of Oikonomidis et al. (2011a) might be the most notable work in this category. However, unlike Stenger et al. (2006) they created the samples at runtime to test the hypothesis and used the power of *graphical processing unit* (GPU) to deal with added computational complexity. Oikonomidis et al. (2011a) used a simple depth discrepancy feature and *particle swarm* (Eberhart & Kennedy 1995; Kennedy et al. 2001) technique to match the proposed hypothesis with observed depth from Kinect.

Search space for the hand configuration is made with 27 DOF. There is a need for four hand samples even to sample two different rotational angles for two phalanges. Hence, joint evidence techniques are computationally expensive than disjoint evidence techniques. However, they deal with occlusion effortlessly. As all the hand parts are drawn together, there is no need for a module to confirm the kinematic constraints.

2.1.8 Disjoint Evidence Techniques

Disjoint evidence techniques (Sudderth et al. 2004b; Hamer et al. 2009; Keskin et al. 2011; Poudel et al. 2013a) consider each hand part separately. Fifteen parts of the five fingers and one palm are sampled and evaluated separately. Each hand part is represented using 6 DOF, 3 DOF for position and

3 DOF for rotation. The number of all DOFs is 96 for disjoint evidence techniques. Even though the number of DOF increased to 96 DOF, tracking hand parts independently decomposes the search space and reduces the search space greatly (Sudderth et al. 2004b).

Sudderth et al. (2004b) introduced the first disjoint evidence technique for 3D hand tracking. Previously, Deutscher et al. (2001) also reported a similar technique for articulated body motion capture. Sudderth et al. (2004b) sampled and evaluated each hand part independently. They used nonparametric belief propagation (Isard 2003; Sudderth et al. 2003) for global optimization of the hand configurations.

Recently, the work of Keskin et al. (2011) might be the most impressive work in this category. They used Kinect (2013) depth sensor for hand pose estimation. Also, the work of Keskin et al. (2011) falls in the appearance based approach. Keskin et al. (2011) used random forest (Breiman 2001) to classify each hand part and means-shift (Comaniciu & Meer 2002) to find the center of the probability mass function i.e. hand joints. They used *neural networks* to predict the occluded joints.

Disjoint evidence techniques are computationally efficient because they reduce the search space. However, these techniques need additional mechanisms to handle the occlusion and hand parts collision.

2.2 Human Body Tracking

Full human body tracking and hand skeleton tracking shares some common problems such as a tree-like connectivity, shape deformation and size variability. Human body tracking research has a long history as early as O'Rourke & Badler (1980) and Hogg (1983). Aggarwal & Cai (1997), Moeslund et al. (2006) and Poppe (2010) reviewed previous work of the human body tracking. This section summarizes the human body tracking works relevant to the hand tracking problems.

In the last decade, most of the computer vision based human motion capture or body tracking techniques treat each limb, head and torso independently (Felzenszwalb & Huttenlocher 2000); such a strategy reduces the search-space and allows softening of the constraints on the joints. Such a model is also known as a *loose-limbed* model (Sigal et al. 2003). Sapp et al. (2011) proposed an ensemble of *stretchable models* for upper body tracking and they represented each hand using a separate stretchable model. The main contribution of Sapp et al. (2011) is joint representation as *unary potential*, which makes their model adaptive for various lengths of limbs.

Leibe et al. (2008) proposed an object detection technique called *implicit shape model* (ISM). They learned visual words to predict the center of the 2D objects. Later, Muller & Arens (2010) applied ISM to body tracking. They learned offsets from the visual words to predict the center of each body part. Gall & Lempitsky (2009) also applied the ISM to body tracking but they replaced visual words learning with a *random forest* (Breiman 2001). They learned the voting offsets in the leaf nodes of the random forest for each body part center. The main advantage of using random forest instead of learning visual words is that even the pixels far from the object/body-part can vote for their center.

Availability of the consumer depth sensor (Asus 2013; Kinect 2013; Primesense 2013) encourages the appearance-based body skeleton tracking research. Ganapathi et al. (2010) used a Swissranger SR4000 *time-of-flight* (Lange & Seitz 2001) camera and tracked the markerless full body. Shotton et al. (2011) proposed the appearance-based body skeleton tracking system using Kinect depth sensor (Kinect 2013). Shotton et al. (2011) used *random forest* (Breiman 2001) for body parts classification and *mean-shift* (Comaniciu & Meer 2002) to collect the body part classification evidence for body joints predictions. The work of Shotton et al. (2011) was very impressive as they tracked bodies with different shapes and sizes in unconstrained environment in real-time. Girshick et al. (2011) improved further using *regression forest* and efficiently predicted all joints including occluded ones. Sun et al. (2012) went a step further by clustering similar poses together before regressing the joints.

The motion of the human body is more predictable and slow, while the hand follows much random path and the motion of the fingers only take a fraction of a second (Tomasi et al. 2003). The shape variation of the human body has greater effect than that of the hand. However, color and texture of clothings provide reliable features for full-body tracking (Ramanan & Forsyth 2003). Hand parts appearances are very similar to each other while the head and torso provides unique cues for the body localization. In most cases, the body is upright but the hand makes random orientation. The number of meaningful hand configurations are much higher than of the body and self-occlusion of the fingers is severe (Keskin et al. 2011). In addition, the size of the fingers are very small compared to the body, which makes fingers less distinctive than body parts.

2.3 Skin Color

Most of computer vision based hand tracking techniques use skin cue to localize the hand region as well as to extract the hand model (Stenger et al. 2006; Sudderth et al. 2004a; Hamer et al. 2010). This thesis has used skin color for hand localization and hand model extraction. Skin detection is a difficult task due to the illumination variation, camera characteristic, ethnicity variation, individual characteristic and other factors.

Skin color detection has two important parts: one is color space selection and another is color modelling. RGB (red, green and blue channels)

(Crowley & Berard 1997; Bergasa et al. 2000; Brown et al. 2001; Jones & Rehg 2002; Sebe et al. 2004), HSV (hue, saturation and value channels) (Huynh-Thu et al. 2002; Wang & Yuan 2001; Zhu et al. 2004), CIE-Lab (international commission on illumination, lightness and a and b color-opponent dimensions) (Cai & Goshtasby 1999; Kawato & Ohya 2002), YCbCr (luma, blue difference and red difference) (Hsu et al. 2002; Wong et al. 2003), and normalized RGB (Brown et al. 2001) are popular color spaces, with RGB and HSV being the most frequently used. CIE-Lab uniformly represents the color based on how two colors differ to the human observer. Modelling brightness also know as light intensity is easier with HSV than RGB. However, most systems choose RGB color space because the illumination variation can be eliminated by increasing the sample size (Jones & Rehg 2002). Due to this advantage, the RGB color space is chosen in most of the hand tracking research.

Skin color modelling falls into three categories: explicitly defined skin region (Peer et al. 2003), non-parametric and parametric methods. The histogram based Bayes classifier is a popular non-parametric modelling approach. Jones & Rehg (2002) used RGB color space and histograms based Bayes classifier and obtained 90% true positive rate with 14.5% false positive rate on unconstrained web images, a dataset made up of approximately 14,000 images. On the parametric skin modelling technique, a mixture of Gaussian has shown the best result (Yang & Ahuja 1999; Terrillon et al. 2000). However, Jones & Rehg (2002) showed that, given enough samples, the histogram based Bayes classifier technique is slightly better than a mixture of Gaussians. Neural Network (Phung et al. 2002), self organizing map (Brown et al. 2001), Bayesian network (Sebe et al. 2004) and a few other methods have been used for skin color modelling.

It is not surprising that skin color detection is a well researched topic. However, most of the work treats skin at pixel level. Skin region is normally made of many pixels together. Hence, viewing skin as region rather than pixel certainly has many advantages. The work of Yang & Ahuja (1998); Kruppa et al. (2002); Jedynak et al. (2003); Sebe et al. (2004) treat skin as group of pixels. Yang & Ahuja (1998) used multi-scale segmentations to find the elliptical region for face detection. Hence, their model is biased toward elliptical shapes. Kruppa et al. (2002) also used a similar concept to find elliptical regions using color and shape information for the purpose of face detection. Sebe et al. (2004) used fixed 3x3 pixel patches to train Bayesian network classifiers using semi-supervised learning, which cannot deform based on the size and shape of the region as the size of the patch is fixed to 3x3 pixels.

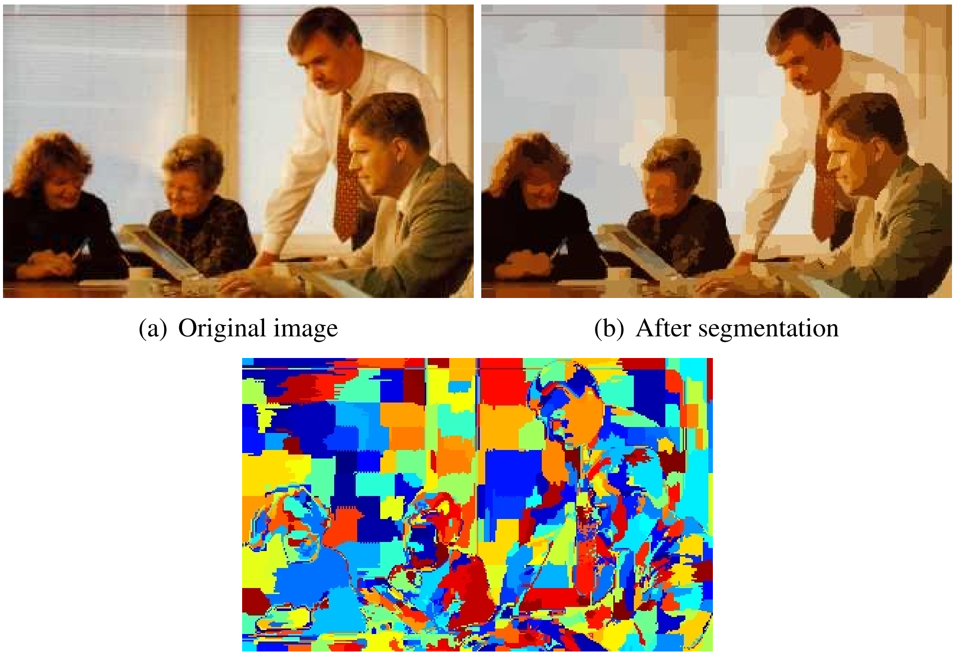
Even skin color of a same person can vary in some extent due to the illumination and background reflections. To tackle with such a problem various adaptive techniques have been proposed. The basic idea of all adaptive technique is changing the pre-learnt color model frequently during the time of tracking. Wu et al. (2000) used an adaptive self-organizing color segmentation algorithm to localize the hand. Stern & Efros (2002) adaptively switched between color spaces to track the face. Zhu et al. (2004) refined a Gaussian mixture model using expectation-maximization during the skin color tracking in videos.

Skin color is an important cue for hand detection and segmentation. However, lighting conditions and skin color variations make the problem harder. Hand tracking techniques which use the skin color cue assume that the user is not using any kind of hand glove and with full sleeve clothing.

2.4 Relevant Techniques

This section describes some of the algorithms and techniques used in this thesis and the rest of the algorithms and techniques are described in the respective chapter's sections only. *Superpixels* and *conditional random fields* algorithms are use in Chapter 3. *Reservoir sampling*, *mean-shift*, and *Markov random fields* algorithms are use in Chapter 5.

2.4.1 Superpixels



(a) Original image

(b) After segmentation

(c) Segmentation visualization

Figure 2.2: *An example of superpixel segmentation.*

The smallest physical unit/point in a digital display device is know as a *pixel* (short form of picture element). Generally, pixels are equidistant in the display devices. However, the number of pixels in each row and column depends upon the display device. The pixels are arranged in a grid structure. During the pixelation process of a scene, the boundary of an object might not be well represented. A region or a collection of pixels is called a *superpixel*, even though there is no hard rule about how to group the pixels together i.e. segment the region. In practice, a five dimensional vector is used to extract the superpixels: three RGB color channels and two positional coordinates of the pixel. The *quick shift* (Vedaldi & Soatto 2008) image segmentation algorithm is one of the popular techniques for superpixel segmentation. Superpixels generated from this approach vary in size and shape, hence the number of superpixels in each image is highly dependent upon the complexity of the image. An image with low color variation will have a smaller number of superpixels than an image with high color variation, as there is no penalty for boundary violation. Generally, the concept of boundary is not used when extracting the superpixels, however different objects have different texture or color which will implicitly act as boundaries. Figure 2.2 on the preceding page shows examples of superpixels of an image. ”The Superpixel extraction library” Vedaldi & Fulkerson (2008) has been used throughout this thesis for superpixel segmentation/extraction.

Recently, Achanta et al. (2012) proposed a simple superpixel segmentation technique. Only one parameter can control the number of superpixels in their technique. The major contributions of Achanta et al. (2012) are speed and memory efficiency for superpixel extraction. Interestingly, their technique can extract approximately same size of superpixels and still preserve the object boundaries. Their technique adopted k-means clustering approach to extract the superpixels. However, there are many other superpixel segmentation techniques exist for examples entropy rate superpixel (Liu et al. 2011), superpixels via pseudo-Boolean optimization (Zhang et al. 2011) and unsupervised segmentation via lossy data compression (Yang et al. 2008).

The popularity of the superpixels is increasing among the computer vision community (Achanta et al. 2012) as it can group similar pixels together and reduce local redundancy. Grouping similar pixels together increases the efficiency for higher level vision tasks such as hand tracking and face detection as the other techniques can operate over superpixels rather than pixel (Poudel et al. 2012).

2.4.2 Reservoir Sampling

Sampling from a stream of data is not a straightforward task when the size of the data is unknown at the beginning, but also sampling from a large stream with known size is a difficult task when all data cannot fit into the computer memory. Even though data may fit in the computer memory, storing the data in the memory and sampling with replacement requires two passes of the data. The problem can be better solved using *reservoir sampling* (Vitter 1985). Reservoir sampling keeps all incoming data stream until the number of the incoming data stream is equal to the sample size. For all incoming stream data, it then replaces the old-sample with probability of sampling size to the current stream data size i.e. probability equal to sample-size/current-stream-data-size.

In most computer vision tasks the size of data is unknown. For example, number of pixels in hand/human-body-part region as number of pixels in body part depends upon the body size, the distance of the body part from the camera and other factors. Girshick et al. (2011) used reservoir sampling to sample the point cloud for *Hough voting* as their technique needed massive memory space and computing power. This thesis also uses reservoir sampling for efficiency reasons. Reservoir sampling is used to collect the votes for joint locations in Chapter 5.

2.4.3 Mean-Shift

Mean-shift is a nonparametric technique for feature space analysis and local mode finding. Originally mean-shift was proposed by Fukunaga & Hostetler (1975) and later popularized by Comaniciu & Meer (2002). Mean-shift is a simple iterative procedure: in each iteration this technique move a mode of the data point toward its optimum point. The procedure terminates when data points stops moving further or other termination criteria are satisfied. Different types of *kernels* can be used with mean shift procedure, which makes some K-mean clustering algorithms a special case of mean-shift (Cheng 1995). The detailed overview of mean-shift can be found in Fukunaga & Hostetler (1975), Cheng (1995), and Comaniciu & Meer (2002).

Mean-shift is a popular technique for multi-model feature space analysis and clustering. Shotton et al. (2011) used mean-shift to find the body joints after calculating the probabilities for each body joint in the whole segmented foreground region. Similarly, Girshick et al. (2011) used mean-shift to cluster and find the center of the Hough voting for body joint detection. Keskin et al. (2011) applied a similar technique for hand joints detection.

Cheng (1995) showed that mean-shift is similar to the gradient descent techniques but it adopts the right gradient step size more effectively. Hence, mean-shift can be viewed as a clustering algorithm along with local mode finding. Mean-shift is one of the efficient mode seeking techniques but it is always stuck in local mode (Comaniciu & Meer 2002). Hence, to find the global mode, a good initialization is necessary, which can be seen as a major shortcoming of the mean-shift technique.

2.4.4 Markov Random Fields

Markov random field (MRF) is a class of graphical model. It is also known as *Markov network* or *undirected graphical model* (Kindermann & Snell 1980). Unlike the *directed graphical model*, the MRF node connections do not have a directional arrow i.e. links between nodes do not carry arrows. Figure 2.3 on the next page shows an example of MRF and details of MRFs can be found in the *Chapter 8* of Bishop (2006).

In the graphical model, the fully connected subset of nodes are called *clique* and denoted by C. The joint distribution over maximum clique is given by (Yedidia et al. 2005):

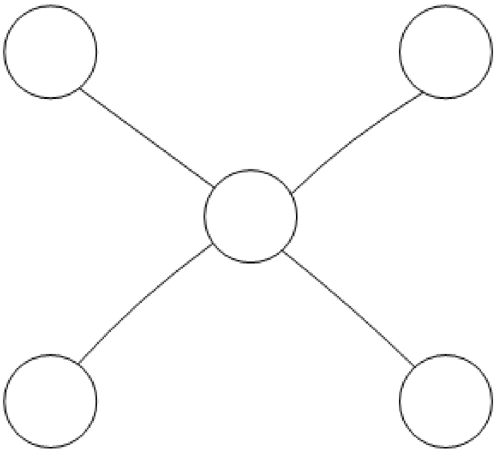


Figure 2.3: *An example of Markov random model.*

$$P(X) = \frac{1}{Z} \prod_{c \epsilon C} \psi_c(X_c) \tag{2.1}$$

where $\psi_c(X_c)$ is a *potential function* and the quantity Z also known as *partition function* is a normalization constant over C is given by:

$$Z = \sum_{X} \prod_{c \epsilon C} \psi_c(X_c) \tag{2.2}$$

Also, $\psi_c(X_c) \geq 0$ is necessary condition to have $P(X) \geq 0$. Any *message passing* techniques can be used to make the inferences in MRF. This thesis uses *generalized belief propagation* technique by Yedidia et al. (2005).

2.4.5 Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty et al. 2001) are a discriminative model, which can optimize the arbitrary *graphical model* known as *undirected graphical model.* CRF offers several advantages over Markovian models and HMM as CRF removes the strong dependence assumptions made by Markov models and HMM. *Maximum entropy Markov models* (MEMMs) and HMMs are biased toward few successor states, while CRF removes such bias

(Lafferty et al. 2001).

This thesis uses CRF for the skin and non-skin region labelling problem. The advantage of CRF in labelling tasks is that CRF optimizes labels globally rather than locally like most of the other techniques such as *conditional Markov model* (CMM) (Ratnaparkhi 1996). Therefore state history is not needed in CRF. Lafferty et al. (2001) showed that CRF significantly outperformed Markov random fields. Hence, in this thesis CRF is used for skin and non-skin labelling tasks in Chapter 3.

CRF has similar structure to that of the conditional Markov model (Ratnaparkhi 1996). It directly models the conditional distribution $P(S|O)$, where S is the state and O is the observed output. CRF allows arbitrary connections and overlapping among nodes unlike HMM and CMM. An example of CRF graph is shown in Figure 2.4. CRF is an undirected graphical model with two layers. One layer describes the state sequence S and the second layer describes the observed output O. In CRF, each output node is connected with every states in state layer. An example of CRF is shown on Figure 2.4.

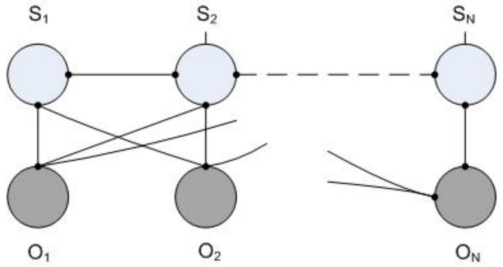


Figure 2.4: *An example of first order CRF graph. The top layer is state sequence and bottom layer shows the output sequence. Output sequence is observable in CRF and is represented by gray circles. In CRF output nodes are connected with every states in the sequence.*

The fully connected subset of nodes is called *clique*. The clique potential $\psi(.)$ maps the label for a given clique with highest positive value among all random variables. It is given by (Lafferty et al. 2001):

$$P(S|O) = \frac{1}{Z(O)} \prod_{c \epsilon C} \psi_c(S_c, O) \tag{2.3}$$

where, $Z(O)$ is global normalization function and given by:

$$Z(O) = \sum_{S} \prod_{c \epsilon C} \psi_c(S_c, O) \tag{2.4}$$

CRF graph can be optimized using any graph optimization techniques. This thesis uses a multi-label graph library called *graph cuts* (Boykov et al. 2001; Boykov & Kolmogorov 2004; Kolmogorov & Zabih 2004) for the inference in CRF graph.

2.5 Summary

This chapter described the research related to 3D hand tracking. Appearance-based techniques are fast but cannot handle the occlusion. Model-based techniques can extract more accurate 3D hand skeleton but are computationally more expensive than appearance-based techniques. Tracking by detection, i.e. hand-pose estimation, is a good strategy to avoid the tracking failure called drifting phenomenon. However, due to the sensor noise recovering a full hand skeleton using a single frame is a difficult task.

Before detection of the hand parts, hand region segmentation is necessary. For the hand region segmentation skin cue is not enough because there are many objects which look similar to the skin color. Hence, skin cue together with depth cue would be more suitable for accurate hand region segmentation. The following Chapter 3 presents the skin color detection and hand region segmentation techniques.

Chapter 3

Skin Cue for Hand Region Segmentation

3.1 Background

Skin color provides a powerful cue in complex computer vision applications such as hand tracking, face detection, and pornography detection. Skin color detection is computationally efficient yet invariant to rotation and scaling. The main challenges of skin color detection are illumination, ethnicity background, make-up, hairstyle, eyeglasses, background color, shadows and motion (Kakumanu et al. 2007). Many of the skin color detection problems can be solved by using *infrared* (Socolinsky et al. 2003) and *spectral imaging* (Pan et al. 2003). However, such systems are expensive as well as cumbersome to implement. Moreover, there are many situations where such systems cannot be used such as image retrieval from the internet.

Most of the skin color detection techniques are *pixel-based* and treat each skin, or non-skin pixel, individually without considering its neighbours. However, it is natural to treat skin or non-skin as regions instead of individual pixels. Hence, this chapter focuses on the *region-based* skin color detection technique for hand region segmentation. Surprisingly, there are only a few region-based skin detection techniques (Yang & Ahuja 1998; Kruppa et al. 2002; Jedynak et al. 2003; Sebe et al. 2004). Kruppa et al. (2002), and Yang & Ahuja (1998) searched for elliptical skin color shape to find the face. Sebe et al. (2004) used fixed 3x3 pixel patches to train a Bayesian network, and Jedynak et al. (2003) smoothed the results using a hidden Markov model. This chapter proposes a new technique purely based on the concept of regions, irrespective of the underlying geometrical shape. Hence, this technique can be easily integrated into any skin detection based system.

The proposed technique uses a segmentation technique called *superpixel* (Moore et al. 2008; Ren & Malik 2003) to group similar color pixels together. Each superpixel is then classified as skin or non-skin by aggregating pixel-based evidence obtained by using a histogram based Bayesian classifier, similar to Jones & Rehg (2002). This technique is also known as a non-parametric technique. However, any suitable pixel-based or superpixel-based skin color classification technique can be used. The result is further improved with *Conditional Random Field* (CRF), which operates over superpixels instead of pixels. Even though the segmentation cost is an overhead in comparison to the pixel-based approach, it effectively reduces the processing cost further down the line such as smoothing with CRF. Aggregation of pixels into regions also helps to reduce local redundancy and the probability of merging unrelated pixels (Soatto 2009). Since superpixels preserve the boundary of the objects (Fulkerson et al. 2009), it helps to achieve accurate object segmentations.

In addition, this chapter presents a region-based skin color detection technique. The work of Yang & Ahuja (1998), Kruppa et al. (2002), Jedynak et al. (2003) and Sebe et al. (2004) are relevant to the proposed technique. However, Yang & Ahuja (1998) used multi-scale segmentations to find elliptical regions for face detection which made their model biased toward skin color elliptical objects. Likewise, Kruppa et al. (2002) also used a similar concept to find elliptical regions using color and shape information for face detection. Whereas, Sebe et al. (2004) used 3x3 fixed size pixel patches for skin detection. The presented technique in this chapter uses patches of varying sizes, which is purely based on image evidence, i.e. skin color in this case. Jedynak et al. (2003) also used a hidden Markov model at pixel level, while this chapter uses conditional random fields and operates on superpixel, as described in Section 3.2.4.

The presented technique not only outperforms the current state-of-the-art pixel-based skin color detection techniques but also extracts larger skin regions and provides semantically more meaningful results while still keeping the false-positive rate low (see Table 3.1 and Figure 3.3 on page 54). This could benefit higher-level vision tasks apart from hand segmentation, such as face and human body detection.

Section 3.2 presents the proposed region-based skin color detection technique; experiments and results are discussed in Section 3.3. Finally, Section 3.4 summarizes the chapter.

3.2 Region-Based Approach

This chapter argues that skin is better presented as regions rather than individual pixels. The proposed region-based approach has four major components: a *basic skin classifier* (Section 3.2.1), extraction of regions called superpixels (Section 3.2.2), superpixels classification (Section 3.2.3), and a smoothing procedure with conditional random fields (CRF) (Section 3.2.4). All components are described in detail in the following sub sections.

3.2.1 Basic Skin Color Classifier

Any good skin color classification method can be used as a basic skin color classifier. This chapter uses the histogram based Bayesian classifier similar to that of Jones & Rehg (2002), a state-of-the-art skin color detection technique.

Learning skin and non-skin histograms: densities of skin and non-skin color *histograms* are learned from the *Compaq dataset* (Jones & Rehg 2002). The Compaq skin color dataset has approximately 4,700 skin images and 9,000 non-skin images collected from free web crawling. 50% skin and 50% non-skin images are chosen randomly for training and the remaining 50% skin and 50% non-skin images are used for testing. The data-set has images from all ethnic groups with uncontrolled illumination and background conditions and the number of manually labelled pixels is nearly 1 billion. Skin and non-skin histograms are obtained in RGB color space with 32 bins for each color channel, similar to the settings in Jones & Rehg (2002).

Skin color classifier: The conditional probability of a color *c* being a skin *s* is given by:

$$P(s|c) = \frac{P(c|s)P(s)}{P(c)} \tag{3.1}$$

where, $P(c|s)$ is the likelihood of a given color *c* being skin, $P(s)$ is skin color prior and $P(c)$ is marginal likelihood of the color *c*. Similarly, the probability of a color being non-skin $\bar{s}$ given a color, *c*, is given by:

$$P(\bar{s}|c) = \frac{P(c|\bar{s})P(\bar{s})}{P(c)} \tag{3.2}$$

where, $P(c|\bar{s})$ is the likelihood of a given color *c* being non-skin and $P(\bar{s})$ is prior for non-skin. Further $P(c)$ could be calculated as:

$$P(c) = P(c|s)P(s) + P(c|\bar{s})P(\bar{s}) \tag{3.3}$$

$P(c|s)$ and $P(c|\bar{s})$ are directly calculated from skin and non-skin histograms. Prior probabilities $P(s)$ and $P(\bar{s})$ can be estimated from the total number of skin and non-skin samples in the training dataset. However, for skin and non-skin classification, comparison of $P(s|c)$ to $P(\bar{s}|c)$ is simply enough. Using Equations (3.1) and (3.2), the ratio of $P(s|c)$ to $P(\bar{s}|c)$ can be simplified to:

$$\frac{P(s|c)}{P(\bar{s}|c)} = \frac{P(c|s)P(s)}{P(c|\bar{s})P(\bar{s})} \tag{3.4}$$

The ratio can be thresholded to produce a skin and non-skin classification rule (3.5). Further, $P(s)$ and $P(\bar{s})$ are also constants and if we assume equal priors, inequality (3.5) can be simplified as (3.6):

$$\frac{P(c|s)P(s)}{P(c|\bar{s})P(\bar{s})} > \Theta \tag{3.5}$$

Therefore:

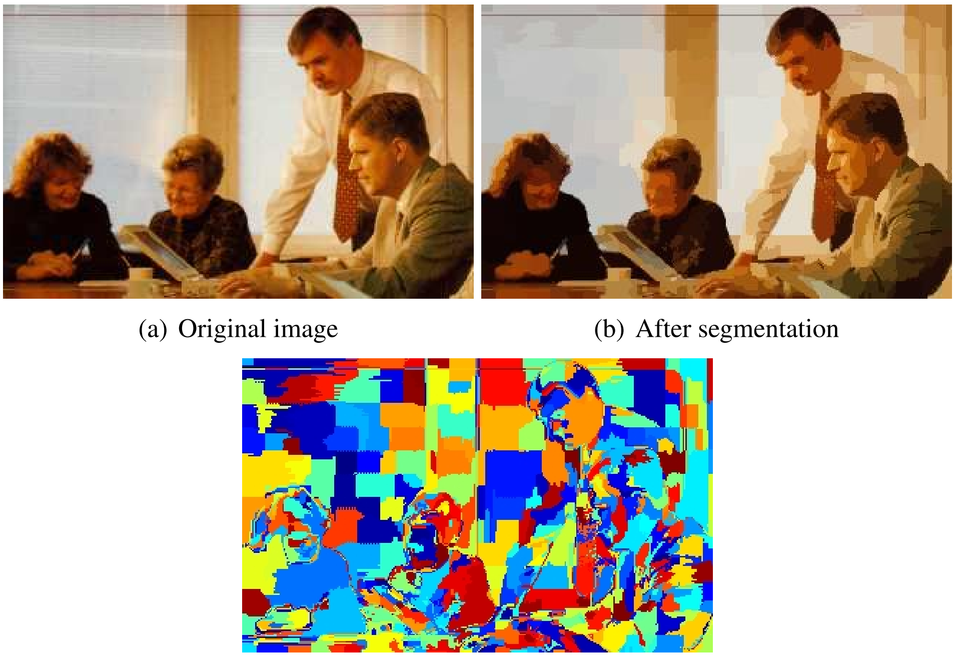
$$\frac{P(c|s)}{P(c|\bar{s})} > \Theta \tag{3.6}$$

where Θ is a constant threshold value for skin and non-skin classification rule.

In the experiments, the values of $P(c|s)$ and $P(c|\bar{s})$ are directly looked-up from normalized skin and non-skin histograms respectively.

3.2.2 Superpixels

A region or collection of pixels is called a superpixel. A five dimensional vector (three RGB color channels and two positional coordinates of the pixel) is used to extract the superpixels, using the *quick shift* (Vedaldi & Soatto 2008) image segmentation algorithm. Superpixels generated from this ap-



(a) Original image

(b) After segmentation

(c) Segmentation visualization

Figure 3.1: *An example of superpixel segmentation. A five dimensional vector is used to extract the superpixels: three RGB color channels and two positional coordinates of the pixel on image.*

proach vary in size and shape, hence the number of superpixels in each image is highly dependent upon the complexity of the given image. An image with low color variation will have a smaller number of superpixels than an image with high color variation, as there is no penalty for object boundary violation. Generally, the concept of boundary is not used when extracting superpixels, however different objects have different textures or colors which will implicitly act as boundaries. Figure 3.1 shows an example of superpixels of an image. This chapter uses ”The Superpixel extraction library” Vedaldi & Fulkerson (2008) for superpixel segmentation. The details of the superpixel technique have been described in Section 2.4.1 of Chapter 2.

3.2.3 Superpixel Classification

First, the pixel based skin color classifier defined in Section 3.2.1 is used to classify the pixels; then the probability of being skin for a given superpixel *sp* with *N* number of color pixels c_i is defined as follows:

$$P(s|sp) = \frac{1}{N}\sum_{i=1}^{N} P(s|c_i) \tag{3.7}$$

Similarly, the probability of being non-skin for a given superpixel *sp* with *N* color pixels c_i is defined as follows:

$$P(\bar{s}|sp) = \frac{1}{N}\sum_{i=1}^{N} P(\bar{s}|c_i) \tag{3.8}$$

3.2.4 Smoothing with CRF

Skin regions have varying size and shape, depending upon the camera angle, and distance between the camera and the human body. Hence, to obtain skin regions, and preserve the skin and non-skin boundaries at the same time, it is necessary to introduce some constraints. *Conditional Random Field* (CRF) provides a natural way of combining pairwise constraints. Color difference and boundary length between adjacent superpixels are used as pairwise constraints similar to Fulkerson et al. (2009). Skin and non-skin labelling *L* of all superpixels *SP* of an image is defined as:

$$-\log(P(L|SP;\omega)) = -\sum_{sp_i \in SP} \Psi(l_i|sp_i) + \omega \sum_{(sp_i,sp_j)\in E} \Phi(c_i, c_j|sp_i, sp_j) \tag{3.9}$$

where ω is the weight of pairwise constraint, *E* is the set of edges of the superpixel, and i and j are node indices of the CRF graph. Each superpixel is represented by a hidden node in CRF graph.

Color potential ($\Psi(l_i|sp_i)$): the color potential Ψ captures the skin and non-skin probability of a superpixel sp_i. We have used skin and non-skin probability for superpixel directly from superpixel classification defined in Section 3.2.3 for color potential Ψ, as follows:

$$\Psi(l_i|sp_i) = \log(P(l_i|sp_i)) \tag{3.10}$$

Edge and boundary potential ($\Phi(c_i, c_j|sp_i, sp_j)$)): pairwise edge and boundary potential Φ similar to Fulkerson et al. (2009) is defined as follows:

$$\Phi(c_i, c_j|sp_i, sp_j) = \left(\frac{B(sp_i, sp_j)}{1 + ||sp_i - sp_j||}\right), [c_i \neq c_j] \tag{3.11}$$

where $B(sp_i, sp_j)$ is the shared boundary length measured in pixel, and $||sp_i - sp_j||$ is the Euclidean norm of the color difference between sp_i and sp_j superpixels.

Only one pairwise potential is used to make the system as simple as possible to show that treating skin color as regions is more effective than pixels. To improve the effectiveness of our skin color detection method, we can add more pairwise potentials similar to those in Shotton et al. (2006). This implementation has only one weighting factor ω, which is optimized using cross validation. We used *max-flow/min-cut* graph optimization algorithm (Boykov & Kolmogorov 2004) for the inference of skin and non-skin regions. The CRF graph is built on the superpixel level, hence CRF optimization is fast (ref. Figure 3.2 on the following page). The details of CRF technique have been described in Section 2.4.5 of Chapter 2.

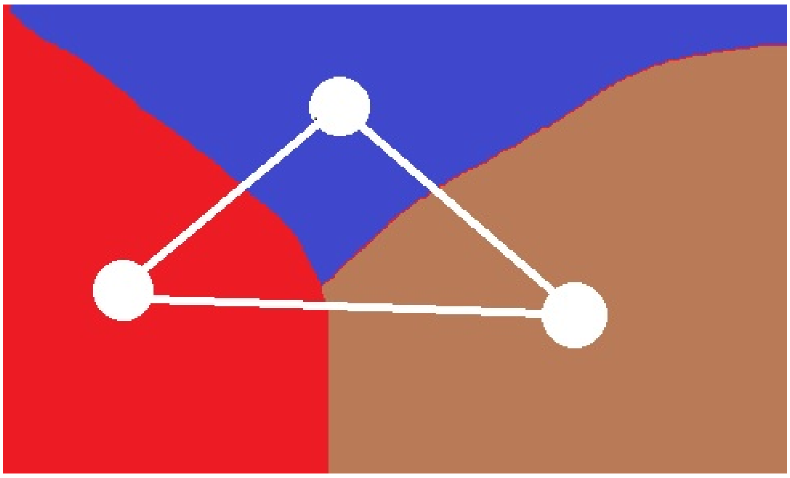


Figure 3.2: *This example picture has three superpixels (red, blue and gray regions). White circles represent the Markov random field (MRF) nodes and white lines represent the connections between two MRF nodes. Even though there are a few hundreds pixels, it has only three MRF nodes as the MRF is built upon the superpixels level.*

3.3 Experiments and results

This thesis uses the Compaq dataset for skin color related techniques. The Compaq dataset has approximately 4,700 skin and 9,000 non-skin images, freely collected from the web. All skin and non-skin images from the Compaq dataset (Jones & Rehg 2002) are divided into two equal numbers of sets, one for training and one for testing. The basic pixel-based skin color classifier mentioned in Section 3.2.1, detects 90% skin color with a 14.2% false positive rate, similar as results found by Jones & Rehg (2002). The bin size of the histogram is equal to $32X3$ (32 for each RGB channel), and threshold constant Θ equal to 1 is used for Equation (3.6).

Superpixel extractions using quick shift are controlled by three parameters: (i) λ controls the trade-off between spatial and color consistency, (ii) σ controls the deviation of the density estimator, and (iii) τ controls the maximum distance in the quick shift tree, which also controls the size of the superpixel. The bigger value of τ produces the bigger superpixels and vice versa. We have used $\sigma = 2$, $\tau = 6$, and $\lambda = 0.9$ for our experiment. These are chosen using a grid search, σ between 0.1 to 5, τ between 1 to 10, and λ between 0 to 1, as there is no explicit mechanism to preserve the skin boundaries. The grid search procedure for an optimum parameter value is briefly described as follows:

1. Initialize search interval- starting value a and ending value b.
2. Initialize number of iteration $iter = 0$ and max iteration $max\_iter = 25$.
3. Initialize number of sampling points $n = 20$, n was constant in our experiments.
4. Initialize error $E_{t=0} = 100$ and change in error $\Delta E = 100$. Error is measured/converted to percentage $\%$.
5. Repeat the following steps until $\Delta E > 0.001$ or number of iteration $iter < max\_iter$.
    (a) E_{t-1} = E_t.
    (b) Uniformly sample n points from interval [a, b] inclusive i.e. interval step $step$ equal to $(b - a)/(n - 1)$.
    (c) Evaluate the error E for all n sampling points/parameter-values.
    (d) Select the lowest error value as E_t and the sample point/parameter-value yielding lowest error as x_t.
    (e) Change $a = x_t - (2 * step)$ and $b = x_t + (2 * step)$.
    (f) Calculate $\Delta E = E_{t-1} - E_t$.
    (g) Increase $iter$ by 1 i.e. $iter = iter + 1$.
    (h) Go to step 4.
6. Return x_t as a best parameter value.

With the above selected parameters it is observed that 97.43$\%$ of skin pixels are correctly grouped into superpixels; i.e. skin pixels which belong to the superpixel and whose number of skin pixels are more than non-skin pixels, with a 0.35% false positive rate. The average size of the superpixels are controlled with the value of τ and σ. Lower values of λ give more importance to the spatial factor while higher values give importance to the color value. Skin color detection depends upon the values of the color channels, hence greater importance is given to the color consistency in superpixel extraction by assigning higher value to the $\lambda = 0.9$. We observed that the skin boundary is not well preserved with higher spatial importance. The average size of a superpixel is 65 pixels in our experiments with huge variation; i.e. size of the superpixel varies from 4 to 400 pixels. However, the size of superpixels is not fixed and fully depends on the complexity of the image.

Method	True Positive	False Positive
Jones and Rehg (2002)	90.00%	14.20%
Proposed technique- superpixel only	**91.44%**	**13.73%**
Proposed technique- superpixel and CRF	**91.17%**	**13.12%**

Table 3.1: *A comparison of results of pixel-based and our region-based techniques.*

Table 3.1 demonstrates the results comparison between the presented region-based technique and the current state-of-the-art pixel-based skin color detection (Jones & Rehg 2002) on unconstrained illumination and background. The region based technique without CRF has 91.44% true positive rate with 13.73% false positive rate, and with CRF whcih has a 91.17% of true positive rate and a 13.12% false positive rate. Simply grouping the pixel-based evidence onto superpixels increased the true positive rate by 1.44% and decreased the false positive rate by 0.48% (ref. Table 3.1). This implies that treating skin as a region yields better results than as pixels. Confusion matrices of our techniques are given on Tables 3.2 and 3.3. Also, confusion matrix of Jones & Rehg (2002) is given on Table 3.4.

The results on Figure 3.3 on page 54 show the effectiveness of the

		Predicted Labels	
		Skin	Non-skin
Actual Class	Skin	033470449	003134234
	Non-skin	051829388	325721862

Table 3.2: *Confusion matrix of proposed technique- superpixel only.*

		Predicted Labels	
		Skin	Non-skin
Actual Class	Skin	033373402	003231281
	Non-skin	049530993	328020257

Table 3.3: *Confusion matrix of proposed technique- superpixel and CRF.*

region-based technique with CRF over the pixel-based method. The region-based technique first groups the skin and non-skin evidence from each pixel into superpixels level using the basic skin color classifier, which helps to remove noise. This is the main reason why only grouping the pixel-based evidence into superpixels increases the true positive rate by 1.44%, and reduces the false positive rate by 0.5% (see Table 3.1). Importantly, CRF further helps to extract larger smooth skin regions by exploiting neighbouring color information and boundary sharing between superpixels.

However, in some cases the region-based technique performs poorly than the pixel-based technique when we apply the CRF. Figure 3.5 on page 55 and Figure 3.4 on page 55 highlight such inaccuracies. Skin-like pixels and high boundary sharing between skin and non-skin regions are the main reason of this failure. Experiment results showed that color difference constraints only perform better when skin regions are very small and narrow. Although, overall CRF with both neighbour color difference and length of boundary sharing constraints performed better. Figure 3.6 on page 56 shows an example

		Predicted Labels	
		Skin	Non-skin
Actual Class	Skin	032944215	003660468
	Non-skin	053612278	323938971

Table 3.4: *Confusion matrix of Jones & Rehg (2002).*

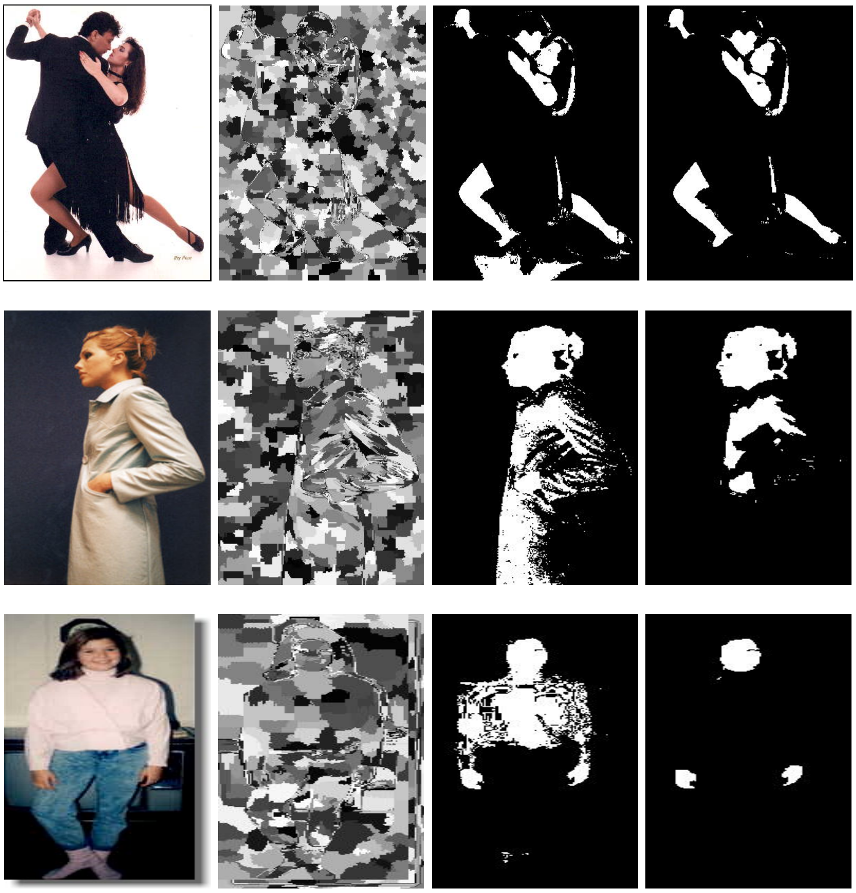
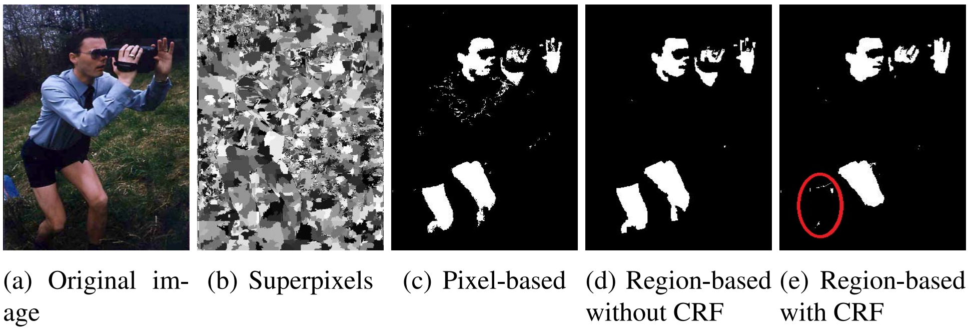


Figure 3.3: *Comparison between pixel-based Jones & Rehg (2002) and region-based skin color classification techniques. The left column shows the original images. The middle-left column shows the superpixels. The middle-right column shows result of pixel-based classification technique and the right column shows the result of region-based classification technique with CRF. CRF helps by exploiting neighbouring color information and boundary sharing between superpixels.*

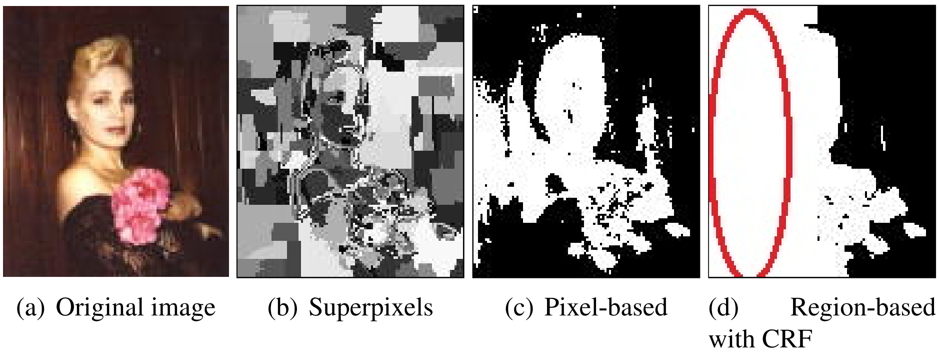
where CRF with both neighbours color difference and length of boundary sharing performs better, than only with neighbours color difference.

Skin regions do not have the same color values, since even the closest skin color pixels within superpixels have different color values, and also other skin-like objects exist. Thus results can be further improved using texture information, which is left for future work.



(a) Original image (b) Superpixels (c) Pixel-based (d) Region-based without CRF (e) Region-based with CRF

Figure 3.4: *This example shows the advantages of the region-based approach even without CRF (see sub-figures c and d). Sub-figures d and e show the failure case without CRF (ref. red color ellipse).*

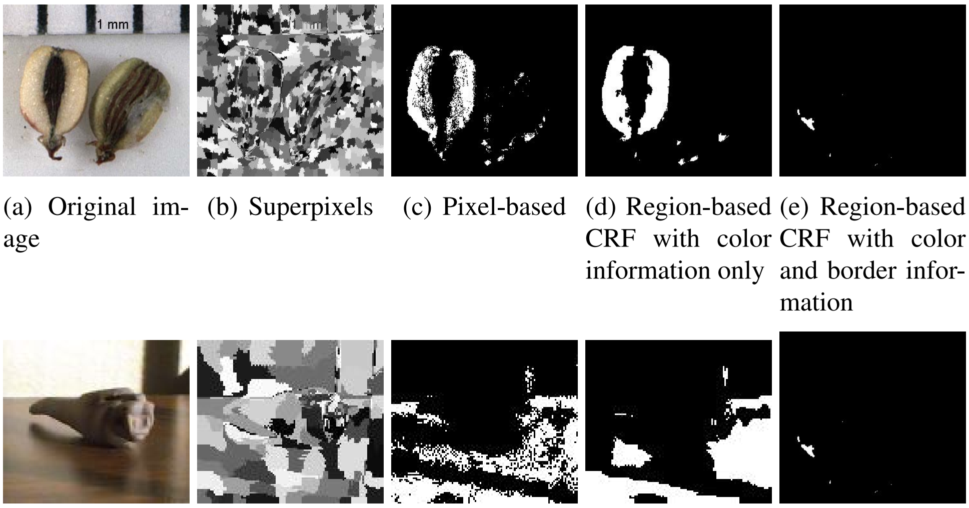


(a) Original image (b) Superpixels (c) Pixel-based (d) Region-based with CRF

Figure 3.5: *This example shows the failure of the region-based approach when border information is applied in CRF smoothing (ref. red color ellipse).*

3.4 Summary

This chapter presented a region-based skin color detection technique, which outperforms the current state-of-the-art pixel-based skin color detection technique Jones & Rehg (2002). The color and spatial distance based clustering technique is used to extract the regions from the images, known as superpixels. In the first step, the proposed technique uses the state-of-the-art non-parametric pixel-based skin color classifier (Jones & Rehg 2002) which is called the basic skin color classifier. The pixel-based skin color evidence is then aggregated to classify the superpixels. Finally, the Conditional Random Field (CRF) is applied to further improve the results. As CRF operates over superpixels, the computational overhead is minimal. However, the proposed



(a) Original image (b) Superpixels (c) Pixel-based (d) Region-based CRF with color information only (e) Region-based CRF with color and border information

(f) Original image (g) Superpixels (h) Pixel-based (i) Region-based CRF with color information only (j) Region-based CRF with color and border information

Figure 3.6: *Example shoeing the failure of a region-based approach when only a color difference constraint is used on CRF optimization.*

region-based technique needed 40-45 milliseconds on a 3.33 GHz Intel processor for a 320x240 image size, whereas the pixel-based technique Jones & Rehg (2002) needed only around 5 milliseconds. However, 80% of added time is required for the superpixel extraction, which could be reduced by GPGPU.

The proposed region-based technique achieved 91.44% true positive rate with a 13.73% false positive rate without CRF optimization, and a 91.17% true positive rate and a 13.12% false positive rate with CRF optimization. Grouping the pixel-based evidence into superpixels increased the true positive rate by 1.44% and reduced the false positive rate by 0.48%. Moreover, the region-based approach produced smoother results than the pixel-based methods.

These results suggest that it is better to use region-based skin color detection technique rather than a pixel-based. By adding more constraints on the CRF similar to Shotton et al. (2006), the detection rate can be improved.

Moreover, any better skin color classification method can be used as a basic skin color classification module, and be easily combined with the proposed region-based skin color detection framework defined in Section 3.2 to further improve the results.

The region based skin detection technique is used in the remaining chapters of this thesis for the hand segmentation.

Chapter 4

3D Hand Tracking using Markov Random Field

3D hand tracking is required to enable complex human-computer interaction but it presents several challenges such as similarity of appearance, high occlusion and complex articulated motion of hand parts. This chapter will focus on a 3D hand tracking solution using multiple cues including skin color, depth, proposed depth-foreground-background (depth-fb) feature and context information. The depth-fb feature measures the discrepancy of the foreground depth and confidence about foreground and background separation; context information utilizes the neighbouring/local information. Further, this chapter presents a palm deformation handling technique and biologically plausible efficient hand parts intersection constraints handling techniques. To the best of our knowledge, the proposed technique is the first that applies context information to improve 3D hand tracking.

4.1 Background

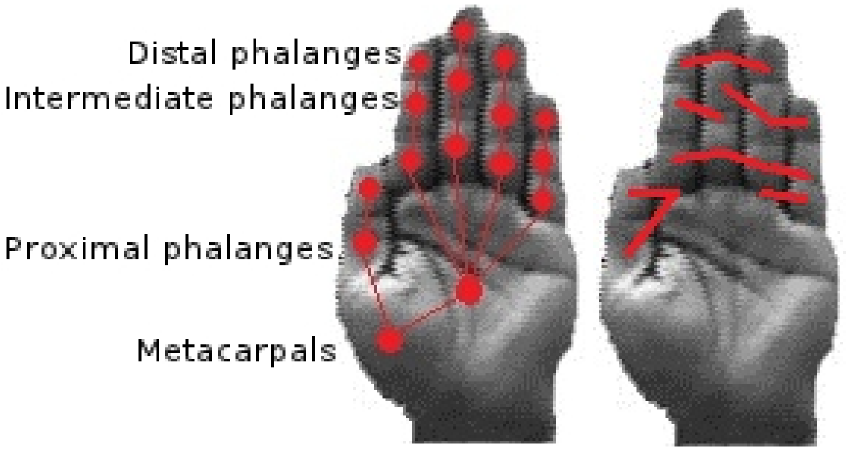
The hand is often considered as one of the most natural and intuitive interaction modalities for human-human interaction (Wang et al. 2007). It is also the most natural interaction interface with the physical world as it is used to manipulate objects by grasping, pushing and twisting (Caridakis et al. 2010). In human-computer interaction (HCI), proper 3D hand tracking is the first step in developing a more intuitive HCI system which can be used in applications such as virtual object manipulation and gaming. Inbuilt cameras in most consumer electronics devices, and the low price of the depth sensors have opened new venues in hand gesture recognition research. However, hand gesture recognition is not a simple task as it requires tracking 27 degrees-of-freedom of hand (ref. Figure 4.1 on page 62) followed by classification of hand postures and movements into meaningful gestures. In effect, the quality of hand gesture recognition is directly dependent on the accuracy of hand tracking. Hand deformation, self-occlusion, appearance similarity and irregular motion are also major problems that make 3D hand tracking a very challenging task. In this chapter, 3D hand tracking is achieved by using multiple cues. Sixteen local trackers have been used, one for each hand part (the palm and 15 phalanges of five fingers; ref. Figure 4.1 on page 62).

This chapter proposes a model-based 3D hand tracking technique. All 16 parts of the hand (a palm and 15 phalanges of five fingers) are sampled and evaluated separately, with 16 local trackers. According to the taxonomy of Oikonomidis et al. (2011a), the proposed technique falls under *disjoint evidence techniques*. The hand is segmented before the evaluation of 3D samples and each hand part sample is evaluated using depth discrepancy features. A new depth-fb feature is also proposed in this chapter. It measures the discrepancy between the foreground depth and confidence about foreground and background separation. The unexplained regions, segmented hand region/pixels which have not been covered by predicted 3D hand model, are used to improve the accuracy of hand skeleton prediction. Since the shape of the palm is highly deformable, a palm deformation module has been proposed to cope with it.

The work of Wu et al. (2005), Stenger et al. (2001), Sudderth et al. (2004b), Stenger et al. (2006), Hamer et al. (2009) and Oikonomidis et al. (2011a) are comparable to the work conducted in this chapter with the following differences. Wu et al. (2005), Stenger et al. (2001), Stenger et al. (2006) and Oikonomidis et al. (2011a) used full hand template matching techniques, while we matched each hand part separately i.e. this chapter we used 16 local trackers (ref. Figure 4.1 on page 62). Treating the hand model in this way, reduces the search space (Sudderth et al. 2004b). Stenger et al. (2001, 2006) also used multiple cameras and *Unscented Kalman filter* (Julier & Uhlmann 1997) for global hand motion estimation, while this chapter uses *iterative closest point* (ICP) (Besl & McKay 1992) for global hand motion tracking. Sudderth et al. (2004b) tracked the hand using a single RGB camera and used edge and color features, whereas this chapter uses a depth sensor. Comparatively, the work of Hamer et al. (2009) is more relevant than others to the proposed work in this chapter. However, Hamer et al. (2009) did not track the palm, which is more deformable than other parts of the hand. Also, the depth sensor used by Hamer et al. (2009) had only 2 millimetres depth error, while the Kinect (2013) sensor used in this thesis has from a few millimetres to about 4 cm depth error (Khoshelham 2011). More importantly, Hamer et al. (2009) did not consider hand-part intersection constraints for the reason that while the hand manipulates an object, fingers do not collide with each other. However, the problem becomes harder when fingers directly collide with each other. The proposed deformation and kinematic correction modules in this chapter deal efficiently with such situations.

The proposed new feature *depth-fb* is robust when finger tips are mostly visible as in Hamer et al. (2009). The major contribution of the proposed technique is the use of context cue in the hand tracking problem. Context cue is used to locate the finger tips, and then ICP is used to correct the position of each distal phalanx by keeping the position of other hand parts fixed, which is called *forward correction*. In the next step, all finger tips/distal-phalanges are kept fixed and other hand parts are searched using the 3D hand tracking technique and it is optimized using the Markov random field (MRF). This step is called *backward correction*. Both steps together are named as *forward-backward correction* in this chapter. Hamze & de Freitas (2004) proposed a fixed structure MRF optimization technique, where half of the nodes were kept fixed and the remaining nodes were optimized. The proposed MRF optimization technique for forward-backward strategy is different from Hamze & de Freitas (2004) because Hamze & de Freitas (2004) enabled or disabled nodes in a predefined pattern, whereas the proposed technique in described this chapter uses context knowledge to enable or disable the nodes in the MRF framework. The context information based technique is more suitable where complex finger movements are required.

The major contributions of this chapter are: i) a depth discrepancy measurement feature called depth-fb, which utilizes edge, foreground and background information (ref. Section 4.3.3); ii) a context cue integration technique for 3D hand tracking (ref. Section 4.9); iii) a palm deformation handling technique (ref. Section 4.8); iv) a hand parts intersection constraints technique (ref. Section 4.5.4); v) The use of a *belief propagation* algorithm in a forward-backward correction scheme (ref. Section 4.9). The remaining sections of this chapter are organized as follows: Section 4.5 describes the 3D hand model; features are detailed in Section 4.3; the hand segmentation technique is detailed in Section 4.4; the 3D hand tracking technique is in Section 4.6 and the results and summary are described in Section 4.10.



(a) MRF model

(b) Positional constraints of MRF model

Figure 4.1: *Both images are taken with a Kinect (2013) RGB-camera (please note that Kinect has a low resolution RGB camera). Hand model: circles/nodes represent hand parts in the Markovian network, and the lines represent pair-wise connections between hand parts in sub-figure (a) to enforce kinematic constraints and (b) shows the positional constraints between hand parts to prevent hand parts intersection in 3D space. Each part of the hand has one local tracker i.e. 16 local trackers (3 phalanges of 5 fingers and one palm). Please note that the hand has 27 degrees-of-freedom (DOF). The palm has 3 positional and 3 rotational DOF; metacarpal has 2 rotational DOF; proximal phalanx has 2 rotational DOF; the intermediate phalanx has 1 rotational DOF; distal phalanx has 1 rotational DOF.*

4.2 Observation Model

A Kinect (2013) depth sensor has been used to capture the data. Kinect (2013) has a RGB/color camera, an infrared (IR) projector, and an IR camera. The depth data is computed using the IR projector and the IR camera. The IR projector casts the dot-pattern IR into the scene and the IR camera captures the reflected IR pattern. The IR patterns are invisible to both the human eye and RGB camera. Kinect (2013) is therefore a family of *structured light depth sensor*. The depth is estimated using the camera calibration technique and relationship between projected and received IR dot-patterns. The details of structured light depth sensor can be found in Geng (2011). OpenNI (2012) data capturing library is used to capture the Kinect data, which gives 640x480 pixels RGB and depth frames. The depth and RGB images are synchronized using a camera calibration technique, which is built-in in the OpenNI (2012) library.

4.3 Hand Features

The proposed technique uses skin color, depth and context cues. All the implemented features are described below, and Table 4.1 on page 65 summaries all used parameters and their selected values.

4.3.1 Skin Color

The proposed technique uses a histogram-based Bayesian classifier from Jones & Rehg (2002) for skin color detection. Densities of skin and non-skin color *histograms* are learned from the *Compaq dataset* (Jones & Rehg 2002). The details of the Compaq dataset have been described in Section 3.2.1 of Chapter 3. Skin and non-skin histograms are obtained in RGB color space with 32 bins for each color channel. As the Compaq dataset has skin images from all ethnic groups and unconstrained backgrounds, the proposed technique can be equally applicable to people from any ethnic background, and lighting and background conditions.

Following, Section 3.2.1 of Chapter 3 the confidence of being a skin color for a pixel $\tilde{o}_i$ of a sample patch $\tilde{O}$ is defined as:

$$R(s_i|\tilde{o}_i) = \frac{P(c|s)}{P(c|\bar{s})} \tag{4.1}$$

and the confidence of being skin for the whole sample patch $\tilde{O}$ is defined as $R(\acute{S}|\tilde{O}) = \frac{1}{N} \sum\limits_{\tilde{o}_i \in \tilde{O} \neq 0} R(s_i|\tilde{o}_i)$, where $\acute{S}$ is the confidence of being skin for a patch $\tilde{O}$ and N is the total number of pixels whose depth value is not zero on the given sample patch $\tilde{O}$. Then, the final confidence of being skin S for the sample patch $\tilde{O}$ is obtained using the following equation:

$$L(S|\tilde{O}) = \frac{1}{1 + e^{-R(\acute{S}|\tilde{O})}} \tag{4.2}$$

Equation 4.2 bounds the confidence values between 0.5 to 1. We can also use *sum of log of likelihood ratios* instead of *sum of likelihood ratios* to compute $R(\acute{S}|\tilde{O})$ and map the confidence values between 0 to 1 in the Equation 4.2. However, we observed that skin color confidence around the finger tips was very low due to the presence of nails. Hence, to minimize the skin color confidence variation among the distal phalanges and other hand parts, we preferred *sum of likelihood ratios* to compute the $R(\acute{S}|\tilde{O})$. Since we used skin color cue along with depth, our proposed technique is still reliable.

4.3.2 Depth

The depth feature for a hand part measures the discrepancy D between the observed/given Kinect depth frame O, and the sample $\tilde{O}$. In another words, discrepancy measures the dissimilarity between depth frame O and the sample $\tilde{O}$. Discrepancy is only measured for the sampled position where depth (z) in the observed frame O is not null. Discrepancy i.e. depth difference d at a pixel i is defined as $d_i = |z_{\tilde{o}} - z_o|$, if the observed depth value z_o at a pixel i is not equal to null otherwise, $d_i = \sqrt{(x_s - \overline{x}_o)^2 + (y_s - \overline{y}_o)^2 + (z_s - \overline{z}_o)^2}$, where $\overline{o}$ is the nearest not null depth pixel from the pixel i at the observed Kinect frame O. The depth value of the background pixels, after hand-region/foreground segmentation are set to null. Some of the foreground/hand-region pixels depth values are also null due to the depth sensor noise. Finally, the total discrepancy value for a sample $\tilde{O}$ is given by $\acute{D} = \frac{1}{N} \sum\limits_{i \in \tilde{o} \neq 0} d_i$, where N is the total number of pixels whose depth value is not null on sample $\tilde{O}$. The depth likelihood value for a sample $\tilde{O}$ is then given by the *Gaussian*

Param	Description	Value	Optimization
σ_d	Depth noise	10	Grid search between 1 to 20
K	Foreground and background separation threshold	5 mm	Visual observation for values between 2 to 8 mm
Θ	Skin and non-skin ratio threshold	0.8	Grid search between 0.1 to 5
δ	Hand part motion differential between two consecutive frames	25	Observation for values between 10 to 40. However, it is dependent upon the experiment videos and we can incorporate the speed of the hand part as well
δ_{max}	Maximum hand part distance in previous frame $t-1$	max d at $t-1$	N/A
δ_{min}	Minimum hand part distance in previous frame $t-1$	min d at $t-1$	N/A

Table 4.1: *Summary of parameters.*

function, which is lower the depth difference $\acute{D}$ higher the probability and vice versa, i.e.:

$$P(D|\tilde{O}) = e^{-(\frac{\acute{D}}{\sigma_d})} \tag{4.3}$$

where σ_d represents the depth noise. We did a grid search for the value of σ_d between 1 to 20 and found that $\sigma_d = 10$ gives the best result.

4.3.3 Depth Foreground-Background

The *depth foreground-background* (depth-fb) feature measures the discrepancy between the foreground depth and confidence in foreground and background separation. The foreground part penalizes the depth discrepancy for foreground (i.e. hand part) and denoted as $depth - f$. The probability for a $depth - f$ feature is given by the Equation (4.3):

$$depth - f = P(D_f|\tilde{O}_f) \tag{4.4}$$

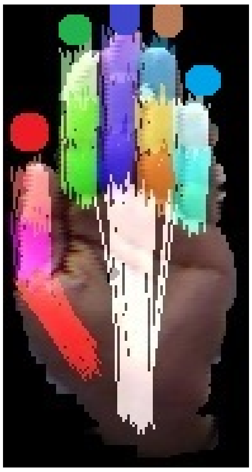


Figure 4.2: *An example of hand part samples. The circles above the finger tips are sample patches for depth-background, depth-b, features, and the samples drawn as lines are centers of the hand parts samples for the depth-foreground, depth-f, feature.*

Even though $depth - f$ and Equation (4.3) follow the same procedure, they are represented separately to distinguish the fact that $depth - f$ is a part of the $depth - fb$ feature.

For the foreground and background separation, i.e. background discrepancy, a small patch is sampled near the finger tip pointing towards the distal phalanx, as shown in Figure 4.2. The patch is about half the length of the distal phalanx with a similar radius. The background discrepancy d at a pixel i in a sampled background $\tilde{O}_b$ is given as follows:

$$di = \begin{cases} 0 \text{ if } |z_{\tilde{o}} - z_o| > K \\ K - |z_{\tilde{o}} - z_o| \text{ otherwise} \end{cases} \tag{4.5}$$

i.e. background discrepancy d at a pixel i is zero if $d_i = |z_{\tilde{o}} - z_o|$ is greater than K, otherwise $d_i = K - |z_{\tilde{o}} - z_o|$, where K is the threshold for minimum distance for foreground and background separation. This is set as 5 millimetres in the experiments.

The total background discrepancy $\acute{D}_b$ for a background patch $\tilde{O}_b$ is given as follows:

$$\acute{D}_b = \frac{1}{N} \sum_{i \in \tilde{O}_b \neq 0} d_i \tag{4.6}$$

where N is the total number of pixels whose depth value is not equal to zero/null in sample $\tilde{O}_b$. The background likelihood, $depth - b$, is given by:

$$P(D_b|\tilde{O}_b) = e^{-(\frac{\acute{D}_b}{\sigma_d})} \tag{4.7}$$

i.e the lower the depth discrepancy then higher the likelihood and vice versa. Thus the total likelihood for being a distal phalanx is $depth - fb = depth - f * depth - b$ i.e.:

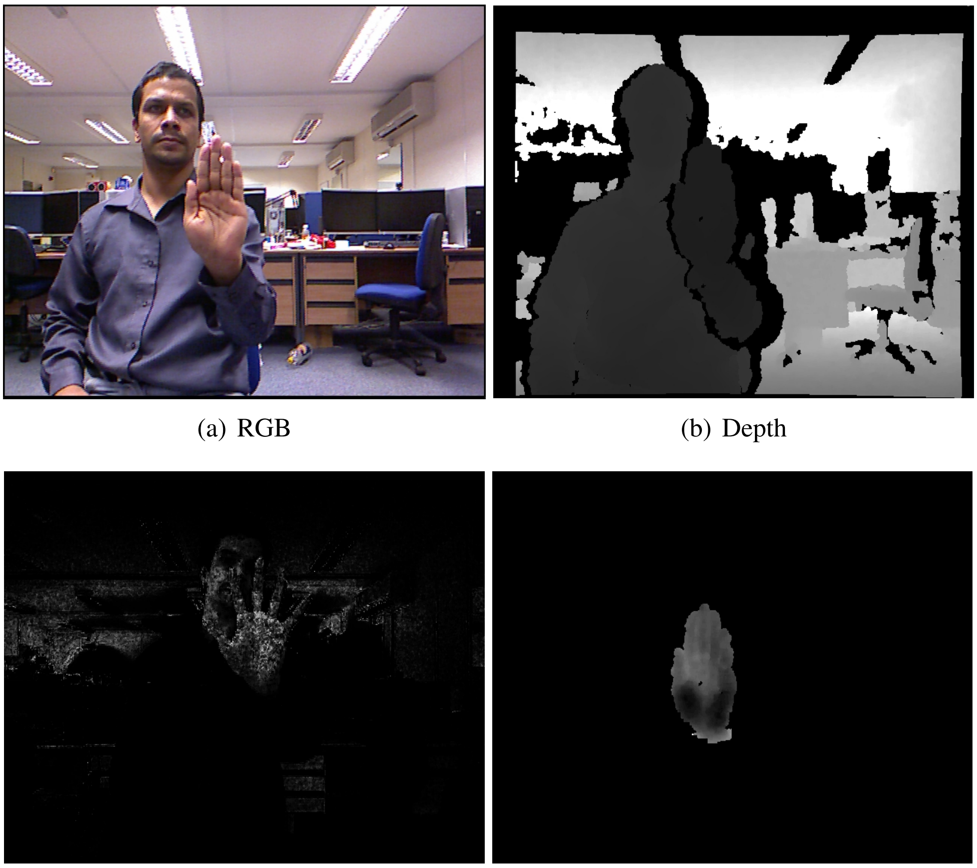
$$P(D_{fb}|\tilde{O}) = P(D_f|\tilde{O}_f) * P(D_b|\tilde{O}_b) \tag{4.8}$$

4.4 Hand Segmentation

Skin and depth cues are used for foreground/hand-region segmentation. This chapter uses a simple thresholding rule for foreground segmentation. The thresholding rule for the foreground fg is defined below:

$$fg = \begin{cases} \text{1 if R(s)} > \Theta \text{ and (d} > (d_{min} - \delta) \text{ and d} < (d_{max} + \delta)) \\ \text{0 otherwise} \end{cases} \tag{4.9}$$

where, $R(s)$ is the skin to non-skin ratio provided by the Equation (4.1), Θ is the threshold value for being skin, d is the depth value and $\delta = 25$ millimetres is the maximum hand parts motion differential between two consecutive frames and d_{min} and d_{max} are respectively the lowest and highest depth values of hand at last frame. It was found that Θ = 0.8 worked better in experiments defined in this chapter. The output of the foreground segmentation is shown in Figure 4.3.



(a) RGB

(b) Depth

(c) Skin likelihood

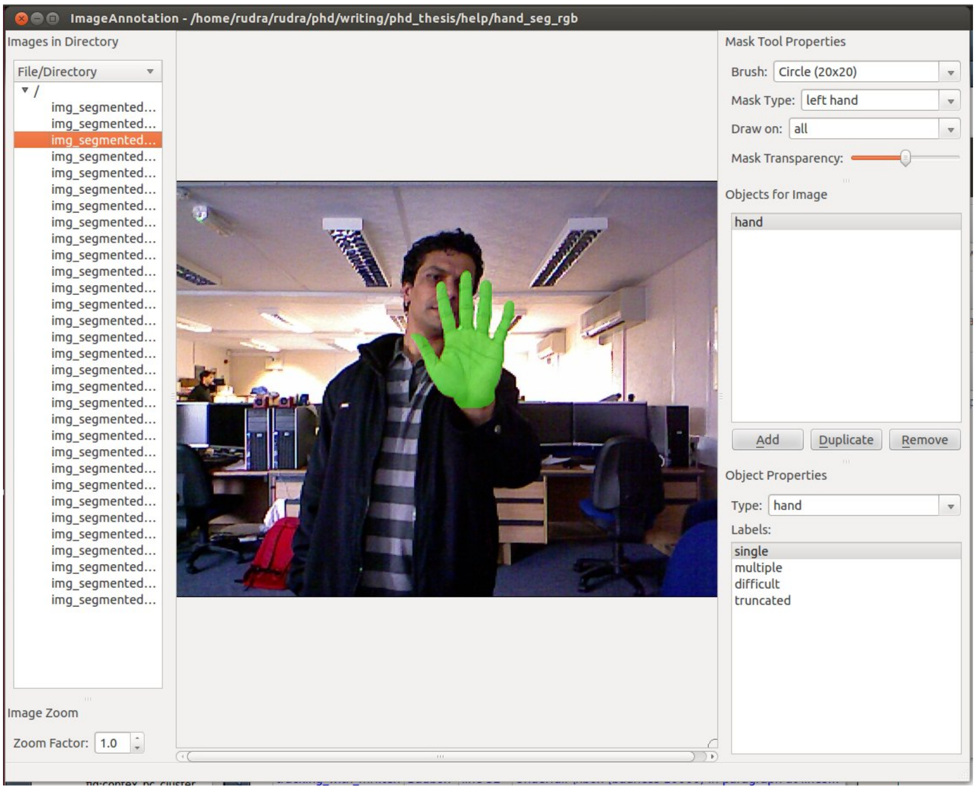
(d) Segmented hand region

Figure 4.3: *a) RGB Kinect image, b) Kinect depth image, c) skin color likelihood and d) depth segmentation output.*

Skin and depth cues help to segment the hand reasonably well. However, this thesis assumes that users use full sleeves. *Jaccard index* (Hamers et al. 1989) has been used to measure the performance of hand region segmentation. It is used to measure similarity between finite sets. The Jaccard index for two sets is defined as:

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{4.10}$$

A is the hand region ground truth, and B is the segmented hand region in our experiment. An *image annotation* application (ref. Figure 4.4 on the


ImageAnnotation - /home/rudra/rudra/phd/writing/phd_thesis/help/hand_seg_rgb

Brush: Circle (20x20)

Mask Type: left hand

Draw on: all

Objects for Image: hand

Type: hand

Labels: single

Zoom Factor: 1.0

Figure 4.4: *A screen shot of hand region annotation application used by this thesis. Green color represents a hand region i.e. ground truth.*

next page) has been developed to annotate the hand region ground truth. An example screen-shot of the hand region annotation application is shown in Figure 4.4..

Only one signer's hand has been used for hand-region segmentation experiment. 13 RGB frames have been manually annotated using the *image annotation* application (ref. Figure 4.4). 13 frames have been selected from two videos. Examples of hand region segmentations are shown in Figure 4.5 on page 71. The average Jaccard index of similarity measurement between the ground truth and segmented hand region was 0.80139 with variance equal to 0.00045 in the experiments. The most of the errors are occurring near hand-sleeve boundary (ref. Figure 4.5 on page 71).

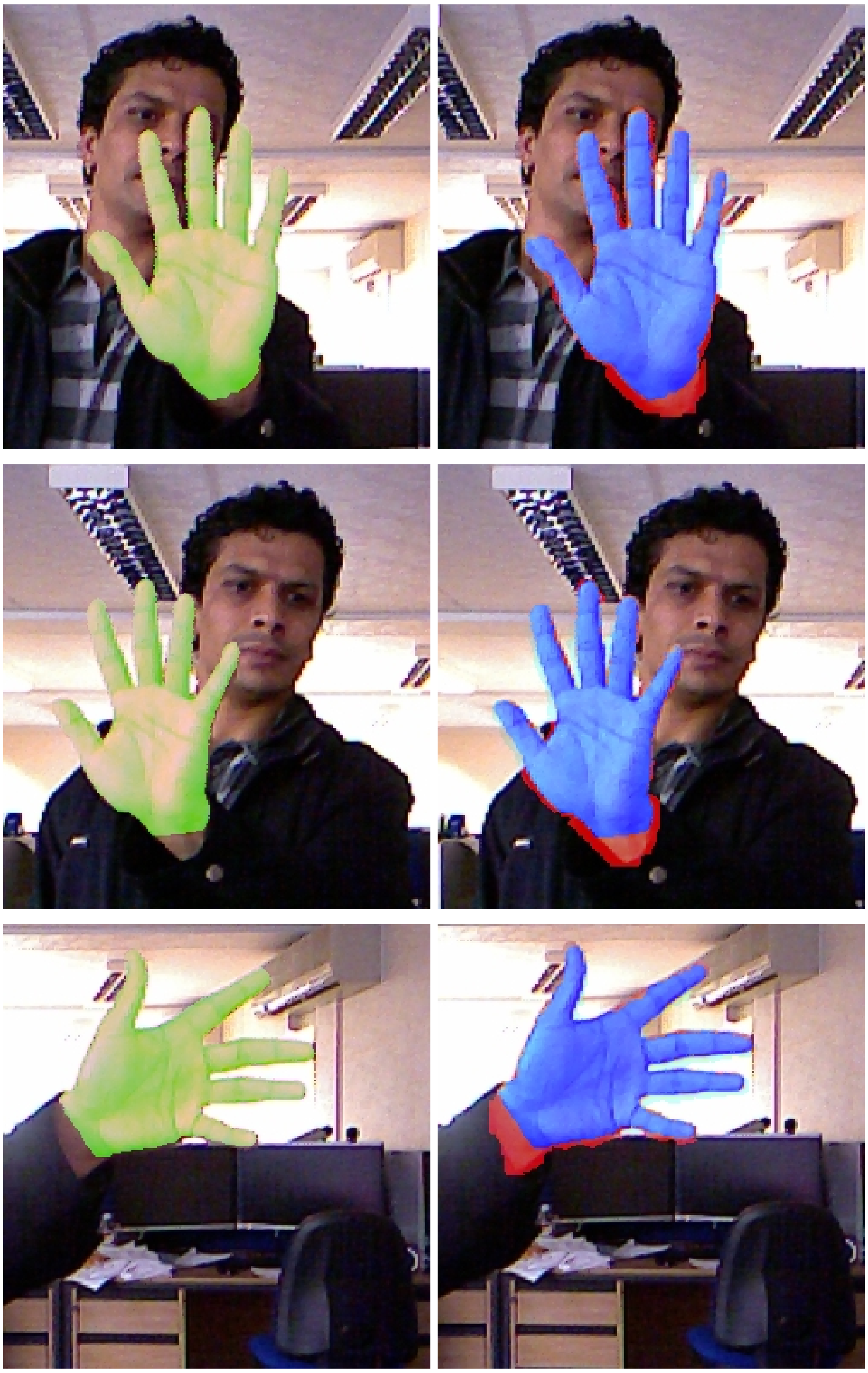


Figure 4.5: *Continue to next page.*

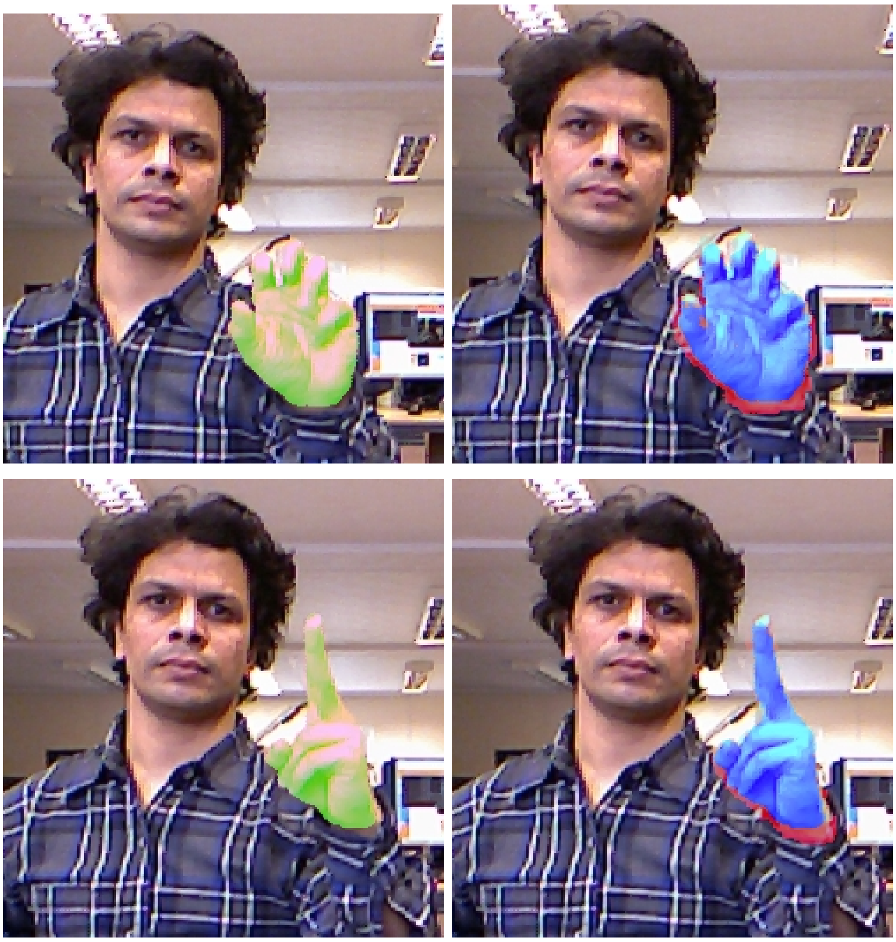


Figure 4.5: *Examples of hand region segmentation.* ***First column:*** *green color (semi-transparent) represents the ground truth.* ***Second column:*** *blue color (semi-transprent) represents the segmented hand region using proposed technique, and red color denotes the segmentation error.*

4.5 3D Hand Model

In this chapter, each hand part is tracked separately then all hand parts are fitted into a Markov random field (MRF) to enforce the anatomic constraints between adjacent hand parts. Individual hand parts are defined as *unary potential* (ref. Section 4.5.2), the kinematic joint between two hand parts is modelled as *pairwise constraints* (ref. Section 4.5.3), and the hand parts intersection constraints are also modelled as *pairwise constraints* (ref. Section 4.5.4) in the MRF framework. The geometrical modelling of the hand parts for sampling is defined in Section 4.5.1.

4.5.1 Geometrical Representation

Unlike the work of Sudderth et al. (2004b), the proposed technique uses a mesh model for the palm because there is more than 15 millimetres of depth variation even in a straight palm surface (ref. Figure 4.3 on page 68). As this chapter has different trackers for each hand part, individual local trackers can settle in different local minima. Hence, improving each local tracker is very important in the proposed technique. Each phalanx is geometrically modelled as a cone, and two spheres are used to fill the cone at both ends (ref. Figure 4.6 on page 75 c and d).

4.5.2 Hand Part Potential

The likelihood of each hand part sample is represented as a unary potential ϕ in the MRF framework. All the unary potentials of an object are represented as a node in the MRF graph. The unary potential ϕ of a hand part is defined as:

$$\phi_i(u_i) = L_s(u_i) * P_d(u_i) \tag{4.11}$$

where L_s and P_d are skin and depth likelihoods as provided by the Equations (4.2) and (4.3) respectively.

4.5.3 Kinematic Constraint

The structural connection between the hand parts is modelled as *kinematic constraint*. It has two sub-constraints- *positional constraint* and *angle constraint*. Thus, kinematic constraints between two hand parts are defined as:

$$\psi_{i,j}^{kc}(u_i, u_j) = P_{pos}(u_i, u_j) * P_{ang}(u_i, u_j) \tag{4.12}$$

Positional constraint makes sure that the two connected hand parts stay joined, and is defined as $P_{pos} = e^{-\frac{E_{pos}}{\sigma_{pos}}}$, where E_{pos} is the degree of positional constraint violation i.e. distance/gap between two connected hand parts sample, and σ_{pos} = 3 mm being the noise factor. The angle constraint is made up from the combination of three sub-constraints- *grasping*, *rotation* and *spreading* as defined below:

$$P_{ang} = P(e^{-\frac{E_{grasp}}{\sigma_{grasp}}}) * P(e^{-\frac{E_{rot}}{\sigma_{rot}}}) * P(e^{-\frac{E_{spread}}{\sigma_{spread}}}) \tag{4.13}$$

similar to the positional constraints, E being degrees of respective angles violation between two connecting hand parts, and σ are normalization factors. This chapter uses $\sigma_{grasp} = \sigma_{rot} = \sigma_{spread} = 10$ degrees, the value for σ is searched between 20 to 1 degree using cross validation.

4.5.4 Hand Parts Intersection Constraint

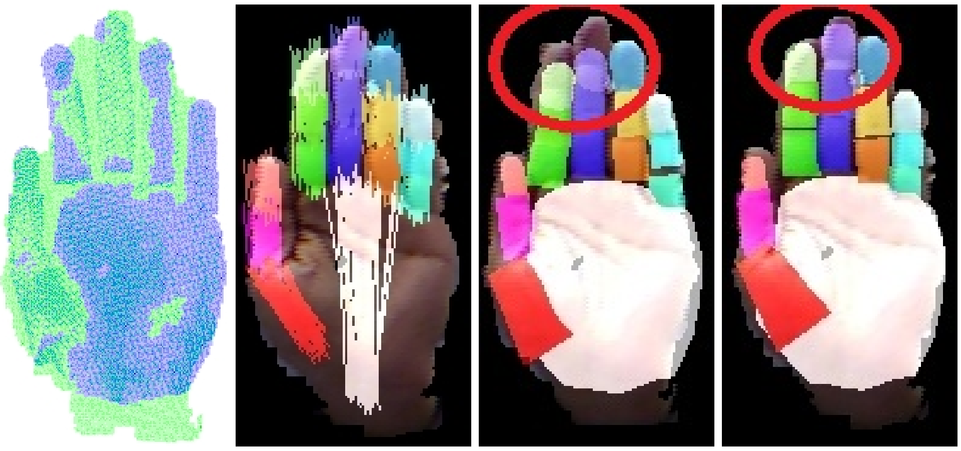
Local hand part trackers do not share any information with each other. Therefore, there are chances that more than one hand part tracker can converge to the same position due to the appearance similarity of the phalanges. Such situations are prevented by using *hand part intersection constraints*. Unlike Sudderth et al. (2004a), intersection constraints used in this chapter do not employ the *volume based intersection* technique, which is computationally expensive. Rather, it uses the Euclidean distance between the closest points (start, mid or end) on the phalanges. The decision to use the closest points among the phalanges is based on the biological structure of the hand, which is shown in the Figure 4.1 on page 62. Intersection constraints penalize the overlapped region among phalanges and are defined below:

$$\psi_{i,j}^{ic}(u_i, u_j) = P_{insec}(u_i, u_j) = e^{-\frac{E_{insec}}{\sigma_{insec}}} \tag{4.14}$$

where E_{insec} is the degree of hand part intersection violation i.e. overlap to each other. If the Euclidean distance between any of the start, mid or end points of two hand parts i and j were less than the sum of their radius, the degree of intersection violation is defined as the sum of the radius of i and j minus Euclidean distance between constraint violated points of i and j in mm. $\sigma_{insec} = 2$ mm is the noise normalization factor.

4.6 3D Hand Tracking

3D hand tracking is tackled in two steps, *global hand motion* tracking, and *local hand parts motion* tracking. The Kinect depth data is converted into a 3D point cloud. An *iterative closest point* (ICP) (Besl & McKay 1992) algorithm is applied on the point cloud data between the observed foreground segmentation at time t_n, and the predicted hand model at time t_{n-1} (ref. Figure 4.6 on the next page a), to estimate the global hand motion. The ICP algorithm iteratively minimizes the distance between two point clouds to estimate the rigid transformation from a source point cloud to a target point cloud. After the global motion transformation, each local hand part tracker samples the parts for local hand part motion/rotation/deformation tracking/estimation.



(a) ICP result

(b) Samples

(c) Edge

(d) Using proposed depth-fb

Figure 4.6: *a) Shows the point clouds of ICP result: green is the observed point cloud at time t_n and the green is point cloud of the predicted hand model at time t_{n-1}. b) Hand parts sampling. c) Output using edge, skin and depth features for distal phalanges. d) Output using novel depth-fb feature for distal phalanges. Depth and skin features are used in all other hand parts except for the distal phalanges. And t is equal to 82 i.e. frame 82.*

All the needed unary potentials, kinematic constraints and intersection constraints are computed from Kinect RGB and depth observation. Finally, all the calculated likelihoods of samples are fitted into the MRF. The MRF is optimized using the *belief propagation* (Yedidia et al. 2005) message passing technique to calculate the marginal probabilities of the local trackers. Similar to Hamer et al. (2009), it has been observed that rather than increasing the number of samples, repeating the local hand parts motion tracking module multiple times with fewer samples is effective, as well as computationally efficient. For example a hand pose estimation yields better results when repeating with 30 sample each times for 5 times, rather than 200 samples for once only. The following two sub-sections describe the ICP for the global hand motion tracking and the message passing technique (Kschischang et al. 2001; Yedidia et al. 2005) for local hand part tracking.

4.6.1 Global Hand Motion Tracking

The 3D hand tracking problem can be divided into two sub-tracking problems namely: global hand motion tracking, and hand parts motion tracking. As all hand parts are connected to each other, they share common global movements and orientation. Further, hand-parts have their own local motions, especially orientations. The hand moves fast in all directions and orientations, hence Gaussian motion tracking algorithms are not suitable. The hand region can be segmented using skin and depth cues (ref. 4.3). In such cases, the iterative closest point (Besl & McKay 1992) technique can be used to estimate the transformation and rotations i.e. six degrees of freedom of the hand in consecutive frames. ICP (Besl & McKay 1992) is the simplest point clouds registration technique. The ICP algorithm converge monotonically to the nearest local minima. However, given an adequate initial position and orientation, ICP can estimate the global optimum solution. This chapter uses the simplest form of ICP and estimates the transformation and orientation using *singular value decomposition* (SVD) (De Lathauwer et al. 1994). The steps of ICP technique cab be briefly described as follows:

1. Target point cloud P_t and source point cloud P_{t-1} are assigned from segmented hand region at time t and predicted 3D hand model at time $t-1$ respectively.

2. Initialize the following parameters: number of maximum iteration (50), maximum correspondence distance (20 cm), euclidean fitness epsilon (0.05 cm) and transformation epsilon (0.5 cm).

3. Repeat the following steps until any of the criteria at (2) satisfy.

    (a) Select the closest sets of points from P_t and P_{t-1} using the nearest neighbour criteria.

    (b) Re-estimate the transformation parameters using the SVD technique.

(c) Stop the ICP if termination criteria are satisfied.

This chapter uses PCL (2012) *open source* library for ICP, which uses a *KD-tree* technique (Bentley 1975) for efficiency.

4.6.2 Local Hand Parts Motion Tracking

The iterative closest point estimates the rigid transformation for global hand motion. Each hand part has different movements, hence separate local trackers are used. The proposed technique uses 16 local trackers: one palm and 15 phalanges of five fingers. For the hand tracking, when a new frame arrives, first ICP estimates the global transformation and that global transformation is applied to each of the hand parts. In the second stage, all hand parts are sampled independently. Finally, the hand part potential (ref. Section 4.5.2), kinematic constraints (ref. Section 4.5.3) and intersection constraints (ref. Section 4.5.4) likelihoods are calculated for all samples.

The message passing algorithm, *belief propagation* (Yedidia et al. 2005), is used to maximize the values defined in Equations 4.11, 4.12 and 4.14 for all samples i.e. to predict the 3D hand model. The message passing algorithm is briefly mentioned below. The details of the algorithm can be found in Yedidia et al. (2005). Sample i with $N(i)$ neighbours, sends a message to the neighbour $j \varepsilon N(i)$ when it gets messages from all neighbouring nodes except j. The message from i to j, $m_{i\to j}(u_j)$, for a sample u_j is defined as:

$$m_{i\to j}(u_j) = \sum \phi_i(u_i).\psi^{kc}_{i,j}(u_i, u_j).\psi^{ic}_{i,j}(u_i, u_j) \prod_{k\varepsilon N(i)\backslash j} m_{k\to i}(u_i) \qquad (4.15)$$

Finally, the belief of a joint proposal is defined as:

$$b_i(u_i) = \phi_i(u_i) \prod_{j\varepsilon N(i)} m_{j\to i}(u_i) \qquad (4.16)$$

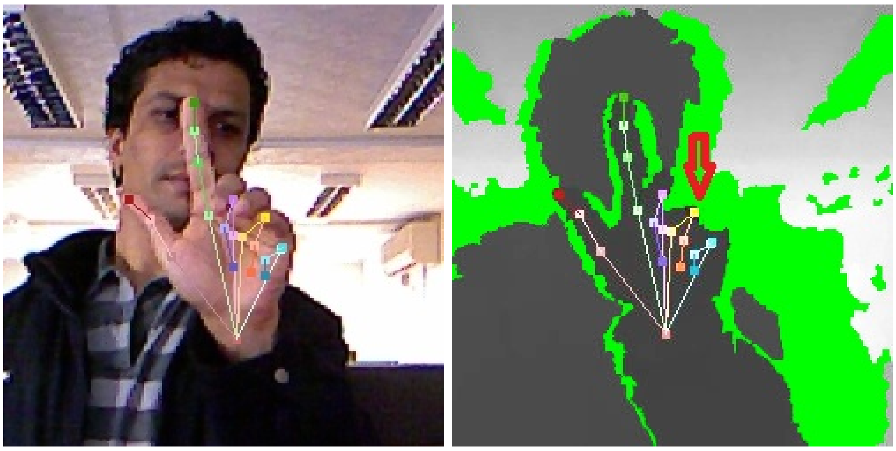
The Gaussian sampling technique is used to sample the hand part. Positional pos standard deviation = 5 cm and rotational $grasp/rot/spread$ standard deviation = 6° are used for Gaussian sampling. Sample size has been 60 x $5 - to - 8 - times$ for each hand part as mentioned in the beginning of Section 4.6. The details of the MRF technique have been described in Section 2.4.4 of Chapter 2. This local hand part motion tracker is repeated three times in our experiments.

4.6.3 Hierarchical Correction

Experiments show that when fingers are overlapped or interlinked together, hand part trackers drift away from the target and after a few iterations *particle degeneracy phenomenon* appears. To overcome these particular problems, the position of each hand part was brought closer to 3 mm whenever the position of a hand part was found to be more than 3 mm from its joint position. The correction has been started by making the palm position fixed and moving the metacarpal phalanx of the thumb within the 3 mm of its joint position in the palm if it has been more than 3 mm distant. Then all proximal phalanges moved within the 3 mm of their respective joint if any of them are more than 3 mm from their respective joint position. Similarly, intermediate then distal phalanges were moved to within 3 mm of their respective joint position. This hierarchical correction proved to be useful in the experiments.

4.7 Tracking with Depth Foreground Background Feature

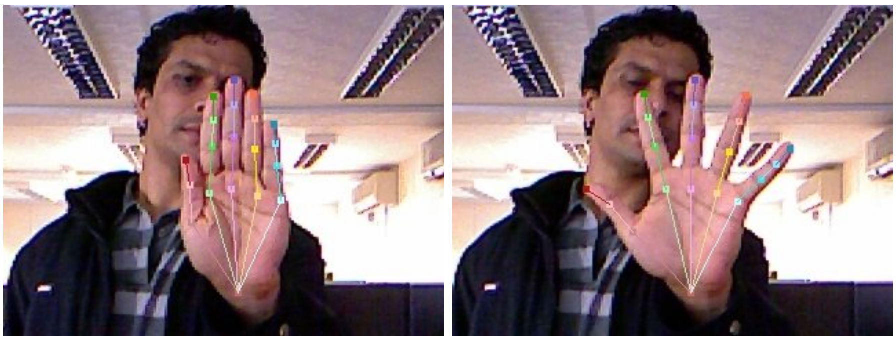
Kinect (2013) depth data is corrupted near the finger edge. This occur especially when the fingers are close to each other, or are close to the background. Figure 4.7 on the following page shows an example of such situations, where



(a) RGB frame #699 (b) Depth frame #699

Figure 4.7: *An example of depth data corruption near the edges. In subfigure (b) middle and ring figure's edges are corrupted.*

depth of middle and ring figure's edges are corrupted. In such a situation tracking with depth feature causes the distal phalanges to move toward the intermediate phalanges. However, Kinect (2013) RGB image does not suffer from the above mentioned problem. Even though both skin and depth information are used, most of the time the distal phalanges tend to move toward the intermediate phalanges. Weak skin probability near the finger tips, due to the fingernail and illumination variation, is a further reasons for the distal phalanges drifting phenomena. To improve tracking, edge probability using the chamfer edge matching technique (Barrow et al. 1977) has been added. In spite of using both skin and depth information together with chamfer edge matching, drifting phenomenon were still occurring, as shown in Figure 4.6 on page 75 due to the edge similarity between distal and intermediate phalanges. To overcome the above mentioned problem, Section 4.3.3 of this chapter proposes a novel *depth-fb* feature, which outperforms the edge and depth together (ref. Figure 4.6 on page 75). The depth-fb feature's background part force distal phalanx closer to near the finger tip. Chamfer edge matching takes 5 milliseconds on average whereas the proposed depth-fb only takes 1.4 milliseconds on single CPU, which is 3.6 times faster on average and is simple to implement.



(a) min state- close hand pose (b) max state- open hand pose

Figure 4.8: *Examples of palm deformation techniques.*

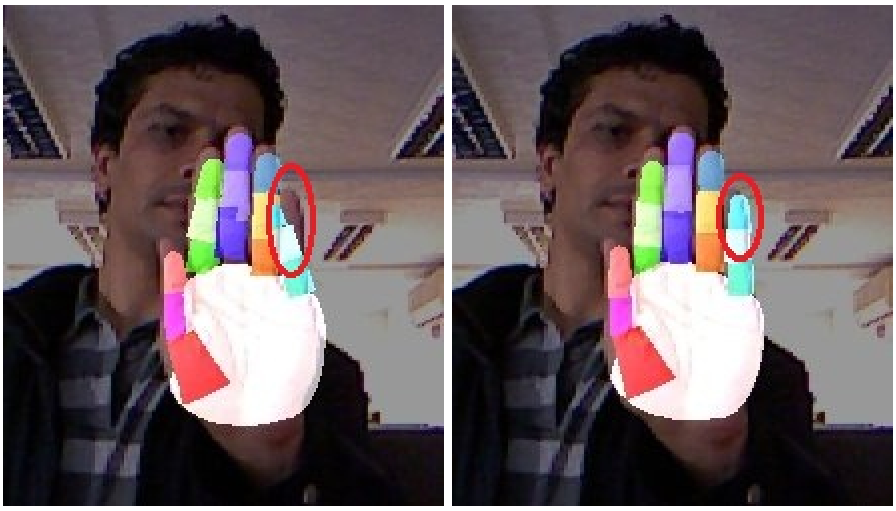
4.8 Handling Palm Deformation

The palm is highly deformable and has a varying surface characteristic. Depth values vary by more than 15 mm due to the non-planarity surface of the palm (ref. Figure 4.3 on page 68 (d)). Hence, discrepancy measurements with elliptical palm model similar to those of Sudderth et al. (2004b), Oikonomidis et al. (2011a) and Stenger et al. (2006) yield a higher discrepancy value, even if the sample position and orientation match correctly. The wrist and elliptical palm model surface look rather similar, which attracts the local tracker of the palm. This makes the problem more challenging. One possible strategy to solve this is by modelling the palm using a mesh model rather than an ellipsoid. However, the palm is highly deformable, as seen through for example spreading to fist poses. To resolve the palm surface deformation issue, the depth discrepancy is measured as follows:

$$d_i = \begin{cases} 0 \text{ if } o_i \neq 0 \\ \sqrt{(x_s - \overline{x_o})^2 + (y_s - \overline{y_o})^2 + (z_s - \overline{z_o})^2} \text{ otherwise} \end{cases} \tag{4.17}$$

and rest of the procedure to calculate the $P(D|\tilde{O})$ follow the Equation (4.3).

The Equation (4.17) ensures that the palm resides within the foreground/ segmented-hand-region, with no penalty for surface variation. To tackle the



(a) Without palm deformation module (b) With palm deformation module

Figure 4.9: *Comparison between without palm deformation and with proposed palm deformation techniques. The red ellipse of sub-figure (a) shows the problem of hand pose estimation without the palm deformation module, and the improved result is marked with the red ellipse of sub-figure (b).*

palm-size deformation, this chapter initialized the hand in two stages: one with a maximum spreading of fingers called *max state*, and one with closed fingers called *min state* (ref. Figure 4.8 on the preceding page). For each proximal phalanx, the spreading angle difference between two states is then equally divided into three intervals namely: *min*, *mid* and *max* deformation states. There can be more than three states depending upon the given test experiment/video. However, only three states performed well in this chapter and we did not experiment with further additional states. Later, these three intervals were used to classify the deformation state of the proximal phalanx. Finally, the most frequent state of the proximal phalanges is used to determine the palm deformation state. The size of the palm and joint positions of the proximal phalanges are changed according to the palm deformation state. Mid-deformation state is defined as the average of the min and max deformation states. This particular deformation technique is applied at 5 frame intervals. Palm size and then proximal phalanges position are assigned to the mid-state for hand tracking without the palm deformation module as most

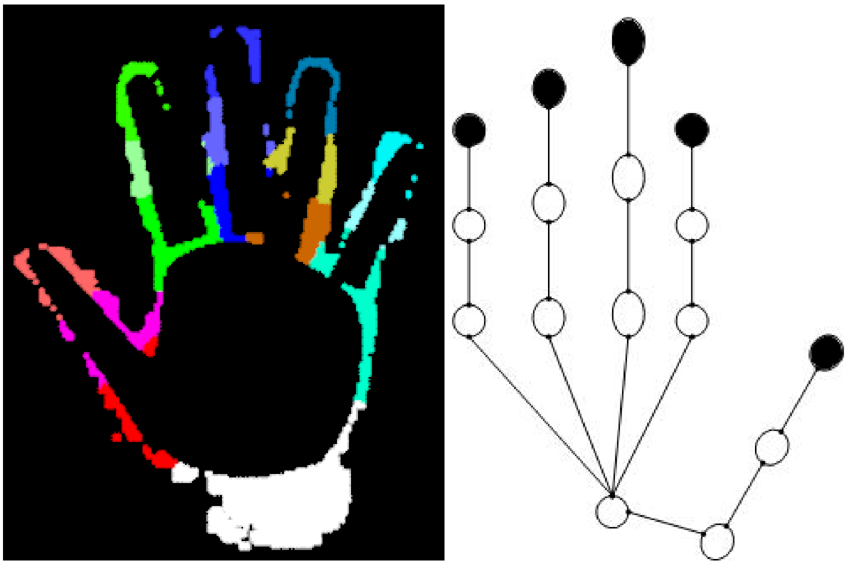


Figure 4.10: ***Left:*** *clustering of unexplained region.* ***Right:*** *black nodes are fixed after finger tips ICP, while white node hand parts are sampled for forward correction. This is used as context information.*

fingers are neither fully spread nor fully closed. Figure 4.9 on the previous page shows the comparison with and without the palm deformation module. As shown in Figure 4.9 on the preceding page, the palm deformation module is able to adapt in palm size variation, while hand tracking without the palm deformation module could not cope with palm deformation, and causes error on joints estimation for the little finger.

4.9 Applying Context Information

As the proposed technique segments reasonably well hand region (ref. Figure 4.5 on page 71), in this chapter we utilize the segmented hand region not covered by the predicted hand model (ref. Figure 4.10), which is named as *unexplained observation*. To cluster the point clouds of the unexplained region, predicted hand part centres are used as fixed centroids and the Euclidean distance as a cost function. The ICP for each distal phalanx is then applied on its clustered regions/point-clouds and overlapped region, if the number of points

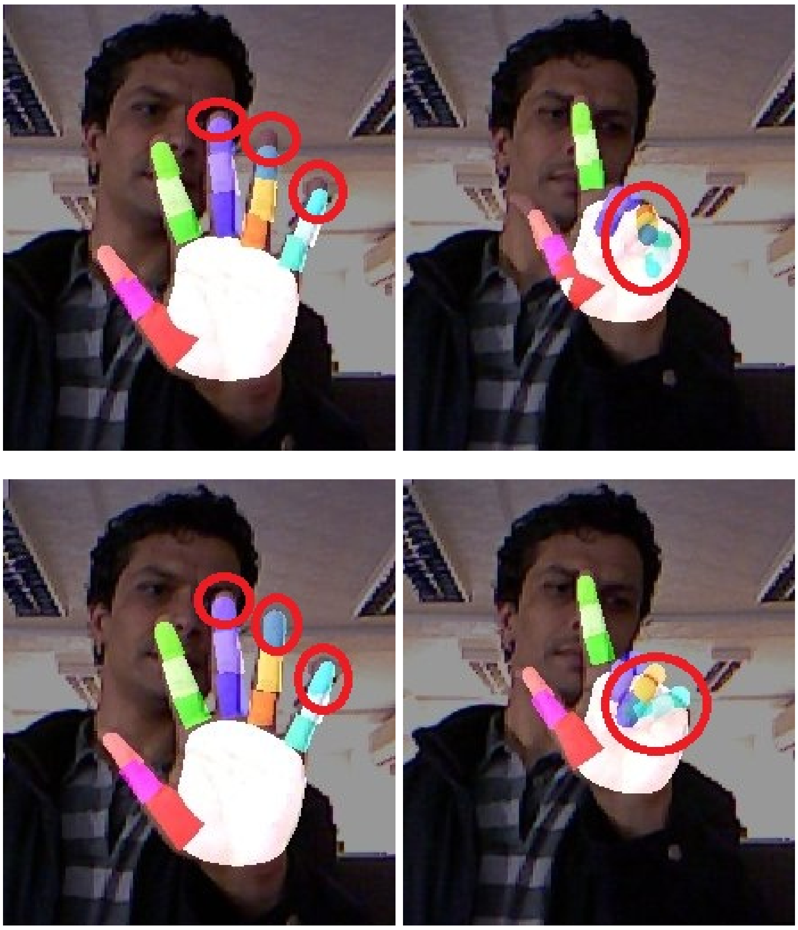


Figure 4.11: *Results comparison between without context cue in first row and with context cue in second row. Index finger and little fingers are tracked well with context cue. Also, the context cue helps to recover the hand pose accurately as shown in second columns.*

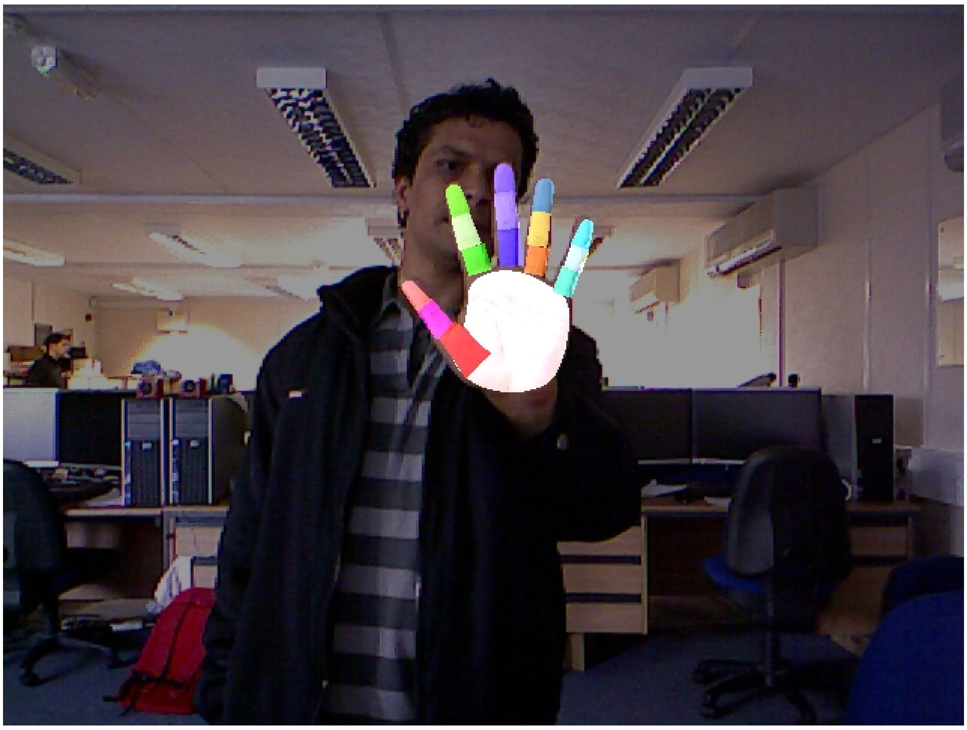
in cluster regions are greater than 20 as shown in Figure 4.10 on the previous page. For the next hierarchical tracking using MRF step, distal phalanges are unchanged and only the other hand parts are sampled (ref. Figure 4.10 on the preceding page).

To our knowledge, this is the first work which uses the unexplained data in 3D tracking, as well as modeling the concept of forward-backward loop correction using the MRF model. Although the work of Hamze & de Freitas (2004) partially enables and disables the nodes to optimize large MRF network in a loop, they followed a predefined fixed structure but the proposed technique in this chapter applies the context cue to decide the fixed nodes. Figure 4.11 on the previous page shows the effectiveness of adding context cue. It works better than depth-fb features, however context cue needed additional computational time. Figure 4.11 on the preceding page shows the comparison between hand tracking using and not using context cue. Figure 4.12 on page 91 shows more results of the proposed 3D hand tracking using context cue.

4.10 Summary

This chapter presented a 3D hand tracking technique using multiple cues: skin color with depth-fb, or depth, and context cue features. The novel depth-fb feature is computationally efficient as it combined the foreground and background information efficiently. The novel context cue feature utilizes unexplained observation and improves the 3D hand tracking. It is efficiently implemented in MRFs network using the forward-backward loop correction technique. The proposed palm deformation technique effectively handled the palm surface deformation, as well as size deformation, and improved the quality of 3D hand tracking. This chapter has presented a biologically plausible hand part intersection constraint, based on euclidean distance rather than on volume intersection technique. The results of hand tracking technique are shown in Figure 4.12 on page 91.

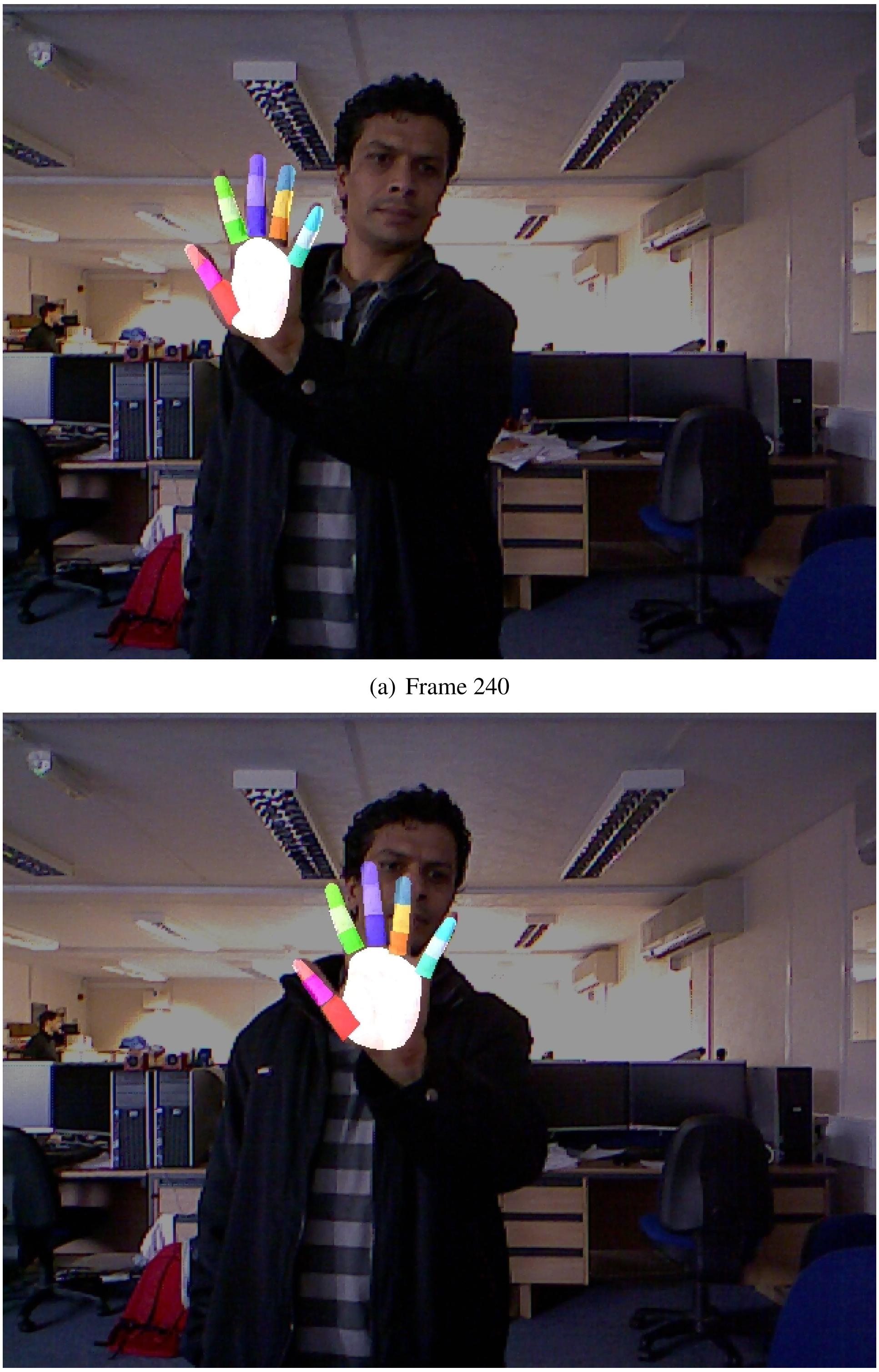
However, the proposed technique has two major drawbacks. Firstly, similar to the *particle filter*, it needs very large sample size (around 300 to 500 for each hard part in each frame) to accurately extract the hand skeleton, which is computationally expensive. To illustrate that, it took 1.8 seconds per frame in a 3.33 GHz Intel processor. Secondly, it required a *hand initialization module* at the beginning of the tracking, and are after tracking failure known as *drifting phenomenon*. These two issues are addressed in the next chapter using a discriminative technique for an example *random forest* (Breiman 2001).



(a) Frame 26

(b) Frame 55

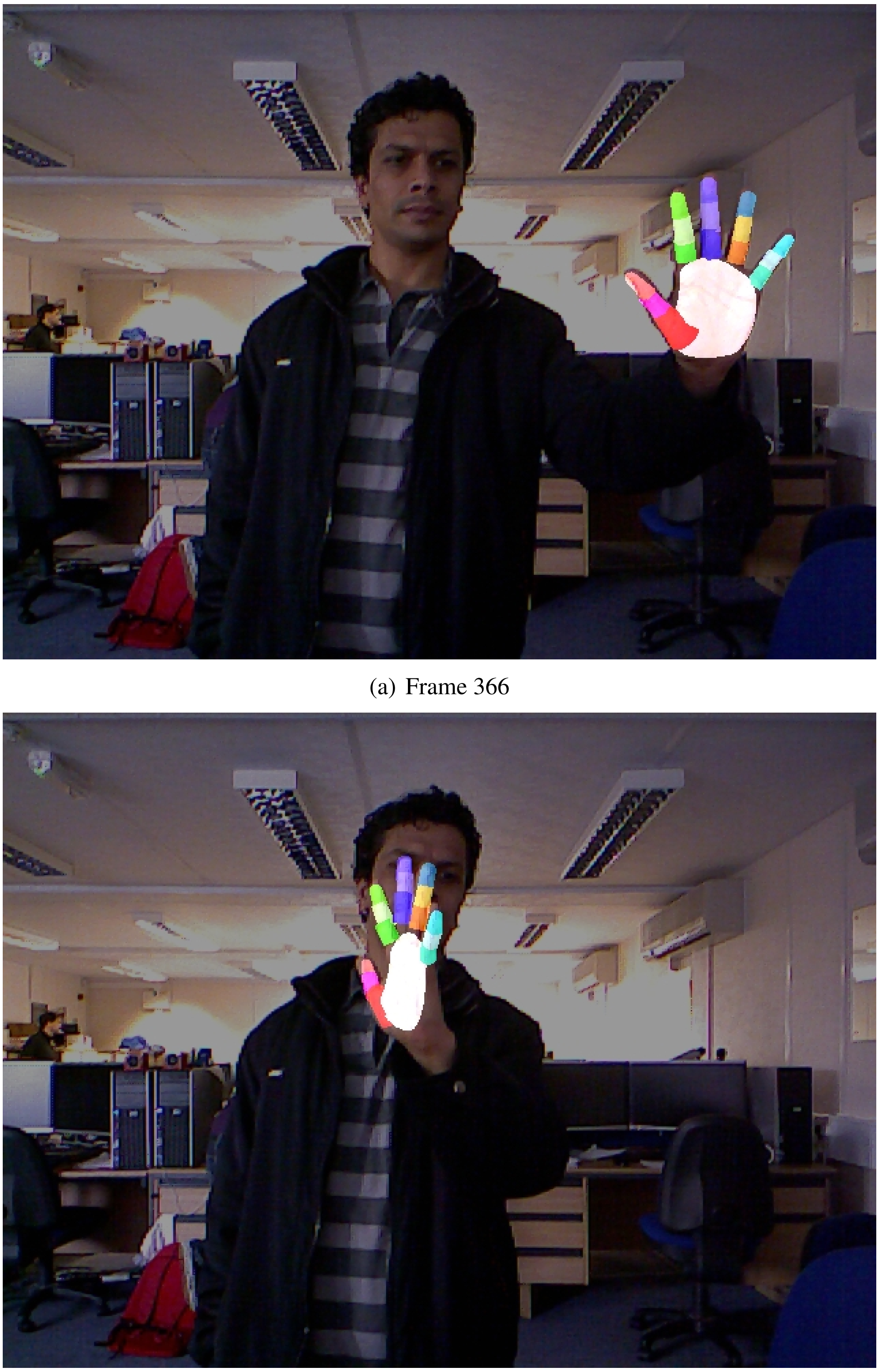
Figure 4.12: *Continue to next page*



(a) Frame 240

(b) Frame 277

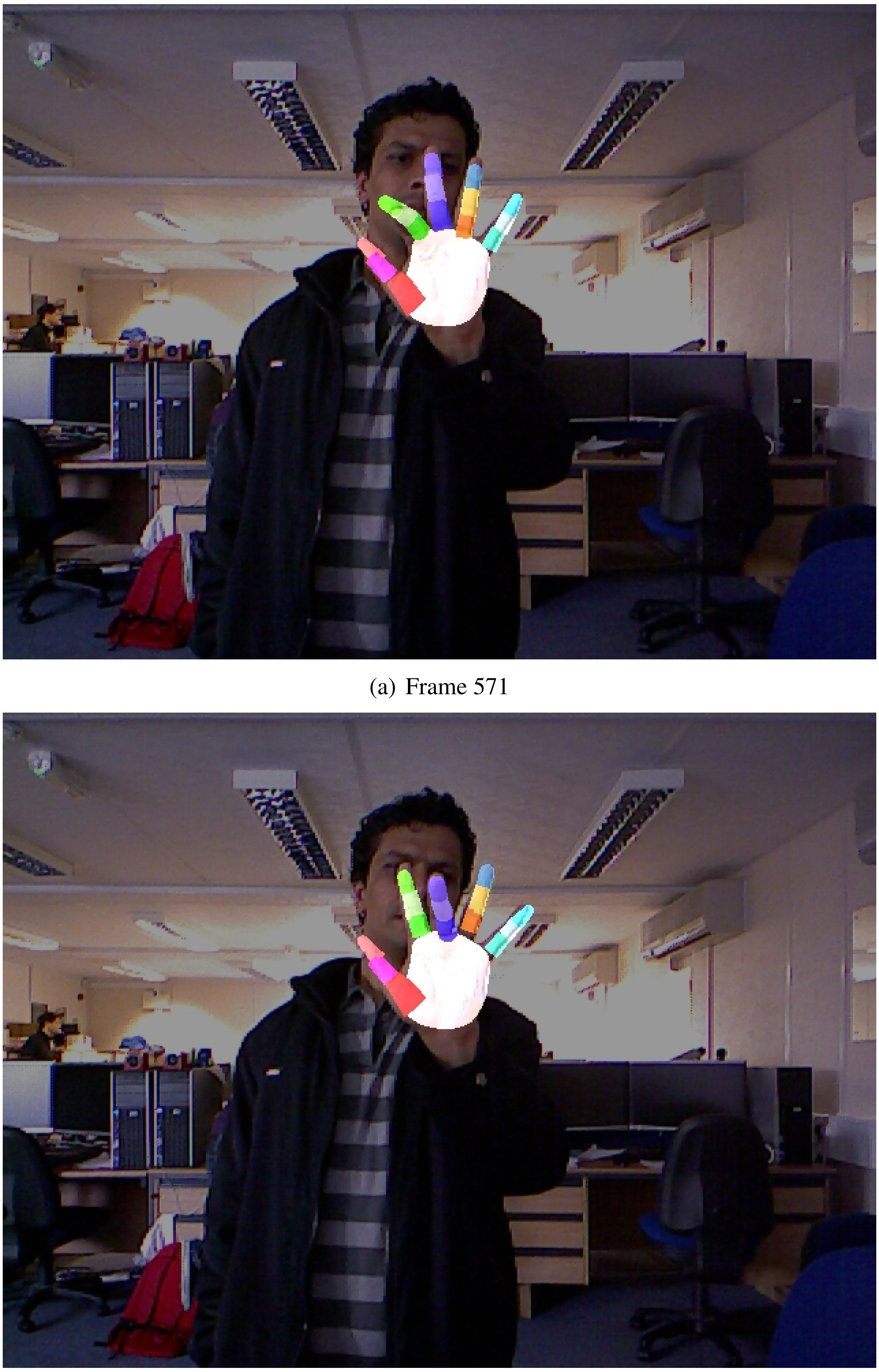
Figure 4.12: *Continue to next page*



(a) Frame 366

(b) Frame 502

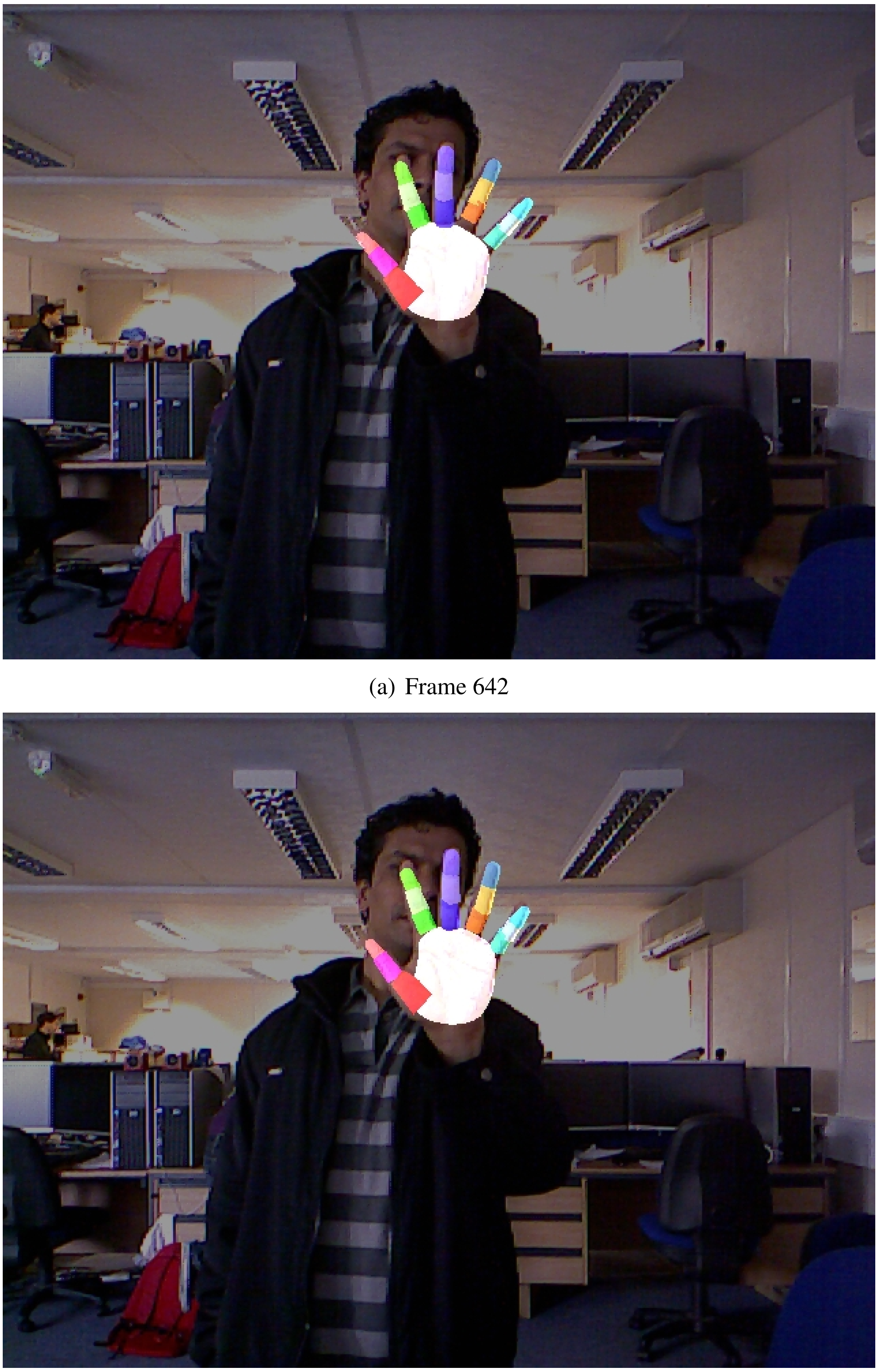
Figure 4.12: *Continue to next page*



(a) Frame 571

(b) Frame 630

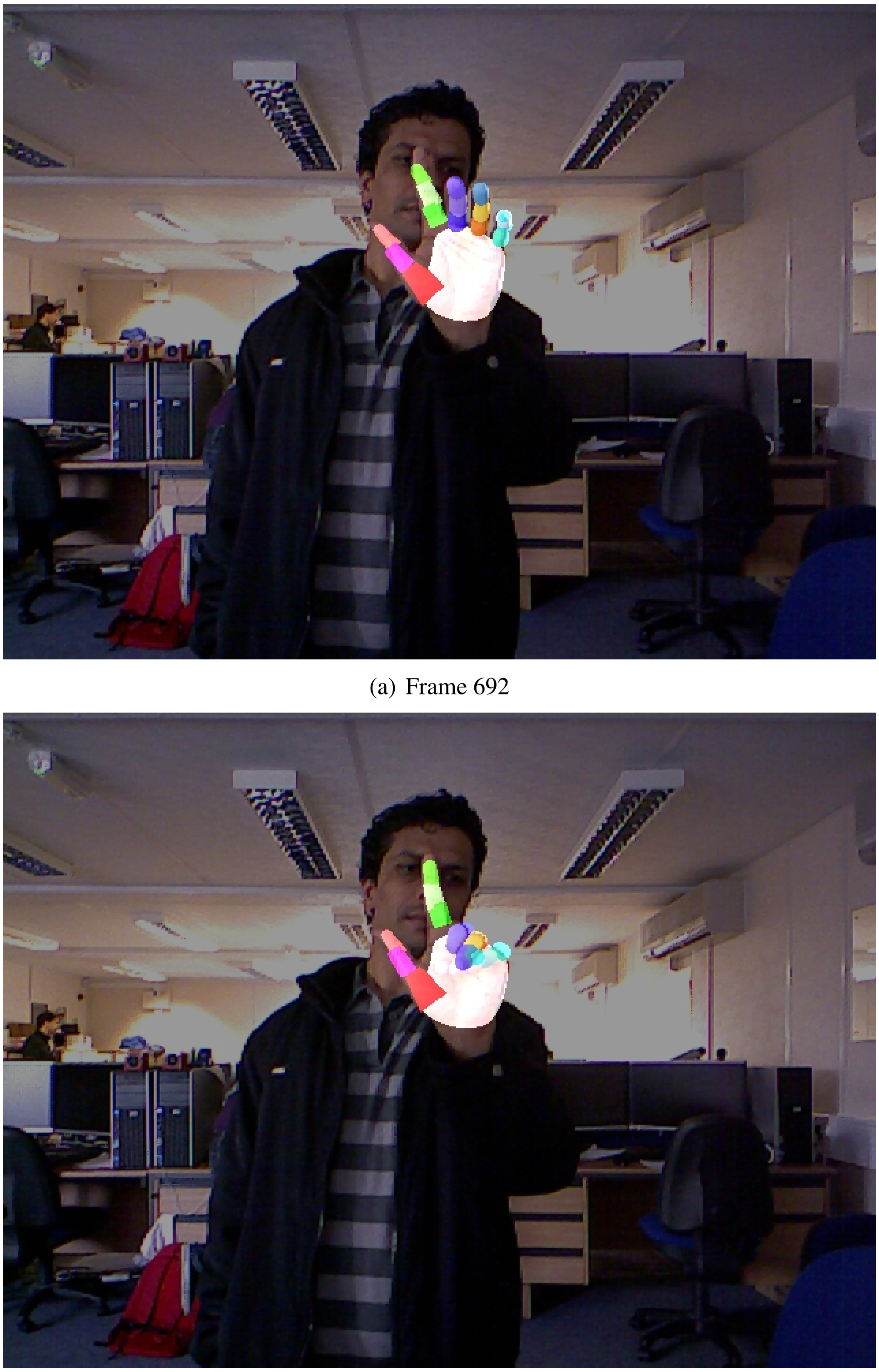
Figure 4.12: *Continue to next page*



(a) Frame 642

(b) Frame 677

Figure 4.12: *Continue to next page*



(a) Frame 692

(b) Frame 705

Figure 4.12: *Results of the proposed technique. It used skin, depth, context cue, hierarchical kinematic correction and palm deformation techniques in the experiment.*

Chapter 5

Combining Discriminative and Descriptive Techniques for 3D Hand Tracking

Discriminative techniques (for example *random forest*) are good for hand part detection, however in other regard they fail due to sensor noise and high inter-finger occlusion. Additionally, these techniques have difficulties in modelling kinematic or temporal constraints. Although model-based descriptive (for example *Markov Random Field*) or generative (for example *Hidden Markov Model*) techniques utilize kinematic and temporal constraints well, they are computationally expensive and hardly recover from tracking failure. This chapter presents a unified framework for 3D hand tracking, using the best of both methodologies. Hand joints are detected using a *regression forest*, which uses an efficient voting technique for joint location prediction. The voting distributions are multi-modal in nature; hence, rather than using the highest scoring mode of the voting distribution for each joint separately, the five highest scoring modes of each joint have been fitted on a tree-structure *Markovian model*, along with kinematic prior and temporal information. Experimentally, it has been observed that relying on a discriminative technique (for example joints detection in case of this thesis) produces better results than a generative technique. Therefore, this observation has been efficiently incorporated in the proposed framework by conditioning 50% low scoring joints modes (here modes of the *mean-shift* in this chapter) with the remaining high scoring joints mode. This strategy reduces the computational cost and produces good results for 3D hand tracking on RGB-D data.

5.1 Background

The unified framework presented in this chapter falls under appearance-based and disjoint evidence techniques. However, the technique in this chapter does not require any additional occlusion or collision handling mechanisms, unlike other disjoint evidence techniques such as Sudderth et al. (2004a); Keskin et al. (2011). The proposed framework consists of three modules: i) hand region segmentation: using skin and depth cues; ii) hand pose estimation: using a regression forest to estimate the positions of the hand joints ; iii) hand tracking: using the pose estimation, kinematic prior and temporal information to track the 3D joints positions.

Inspired by the work of Girshick et al. (2011) which used a *regression forest* to efficiently predict occluded human body joints, the joint estimation module in this chapter uses a discriminative *random forest* (Breiman 2001) to classify the hand-parts and learn joint offsets at leaf nodes. Since the voted joint offsets are multi-modal in nature, a *mean-shift* (Comaniciu & Meer 2002) voting aggregation technique is used. Unlike Girshick et al. (2011), which selects human body joint proposals independently, in this chapter joint proposals with kinematic prior and temporal constraints are optimized globally with a *Markov random field* (MRF) (Yedidia et al. 2005). Temporal information is added on the same semantic level and modelled as MRF (ref. Figure 5.1 on page 95).

The proposed 3D hand tracking technique in Chapter 4 has two major drawbacks. Firstly, similar to the *particle filter*, it needs lots of samples to accurately extract the hand skeleton, which is computationally expensive. The technique proposed in Chapter 4 needed 1.8 seconds per frame in a 3.33 GHz Intel processor. Secondly, the proposed technique in Chapter 4 needed a *hand initialization module* at the beginning of the tracking and after tracking failure, which is known as *drifting phenomenon*. The proposed technique has overcome those two problems.

Keskin et al. (2011) and Hamer et al. (2009) are relevant to the proposed 3D hand tracking technique in this chapter. While Keskin et al. (2011) used a *classification forest* to classify hand-parts, an *artificial neural network* for occlusion handling and *translation vector* to push joints from the finger surface to their inside positions; the proposed technique in this chapter directly predicts the hand joints without requiring an extra occlusion handling mechanism. Moreover, Keskin et al. (2011) track the hand by detection (pose estimation), while the proposed technique incorporates temporal motion and hand-part length prior. On the other hand, Hamer et al. (2009) used a model based approach, whereas the proposed technique uses an appearance based approach. In the MRF model, joints represent MRF nodes, while hand-parts represent MRF nodes in Hamer et al. (2009). Hence the 3D hand tracking technique proposed in this chapter is more flexible for different hand sizes as joint detection is less dependent on the length of the hand parts. Additionally, half of the nodes in our MRF model are fixed, as explained in Section 5.3.3.

The focus of this chapter is on single hand tracking using a Kinect (2013) sensor. The contributions of this chapter are as follows: i) a unified framework for 3D hand tracking which efficiently combines discriminative and descriptive techniques; ii) a regression forest based technique for hand pose estimation which performs better than classification forest based techniques; iii) a simple way of selecting better features from a larger feature space/pool (ref. *feature pool*, Section 5.4).

This chapter is organised as follows: Section 5.2 describes how artificial training data is generated; whilst Section 5.3 presents the proposed 3D hand-tracking framework. Experiments and results are presented and discussed in Section 5.4. And, Section 5.5 summarises this chapter.

5.2 Artificial Data Generation

This chapter aims to build a markerless 3D hand tracking system using a RGB-D sensor. The system is to be trained to detect the hand joints position in an RGB-D image stream. Preparing a real dataset of all possible hand poses with different sizes is almost impossible and time consuming. Therefore to overcome such a problem various computer generated CG hands were used. The trained system is expected to generalize and work equally well on real data. To simulate the Kinect noise, Gaussian noise was added to the CG generated data, which is defined as:

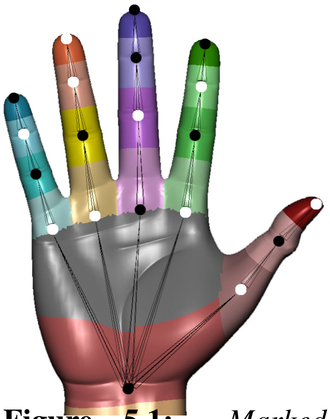


Figure 5.1: *Marked 21 hand regions and MRF model, where white nodes/joints are conditioned on black nodes/joints/fixed-nodes.*

$$d' = d + N(0, \sigma) \tag{5.1}$$

where d is a depth value, d' is a new depth value and $N(0, \sigma)$ is a normal distribution with mean 0 and variance σ. Following the technical details of Kinect published by ROS (2013), the value of σ has been defined as follows:

$$\sigma = \begin{cases} \text{0.001 if d} < \text{0.51} \\ \text{0.001 + ((0.049/4.5) * (d-0.5)) otherwise} \end{cases} \tag{5.2}$$

note: all values are in meters, including depth d. The above equation implements the observation by ROS (2013) that Kinect noise is around +/- 1 millimetre for objects closer than half a meter, and +/- 5 centimetre for objects at 5 meters (m). Hence, $\sigma = 0.001$ m has been used when d is less than 0.51 m, and 0.049 m (which is 0.5 m for object at 5 meters minus 0.001 till distance 0.5 m) is mapped equally between 0.51 m to 5 m.

The system is trained to detect 15 joints of the hand (palm's one, thumb's two and 12 joints of the remaining four fingers) and 5 finger tips, as depicted in Figure 5.1 on the preceding page. Similar to the work of Shotton et al. (2011) and Keskin et al. (2011) the classification forest is trained on 21 regions of the hand; 15 regions are centred around each hand joint and 5 regions for finger tips and one extra region to cover up the middle part of the palm as shown in Figure 5.1 on the previous page. Half a million images were used for the experiments in this chapter, 450,000 images were used for training and 50,000 for testing. This chapter uses artificial data for training and quantitative evaluation of the proposed technique, and the remaining experiments use the real data.

5.3 3D Hand Tracking

The proposed 3D hand tracking framework has three sub modules: *hand region segmentation*, *hand pose estimation* and *hand tracking*. The input to the proposed framework is a stream of RGB-D images. The hand region segmentation module takes both RGB and depth images as input, while the hand pose estimation module takes only segmented depth image as an input. The final hand tracking module takes five high scoring modes of each joint. All the three modules are described in detail below.

5.3.1 Hand Region Segmentation

Artificial data are used for training and quantitative evaluation purpose only, whereas the remaining chapter considered the scenario of the basis of real data for all other experiments. Both skin color and depth cues are used for the segmentation of the hand.

Skin cue: A histogram based Bayesian classifier (Jones & Rehg 2002) is used for skin color detection. Densities of skin and non-skin color *histograms* are learned from the 14 thousand images of *Compaq dataset* (Jones & Rehg 2002) which contains images from all ethnic groups with uncontrolled illumination and background conditions. Training using such a huge dataset makes the proposed technique equally applicable to unconstrained backgrounds, ethnic origins and lighting conditions. The details of skin color detection technique are described in Section 3.2.1 of the Chapter 3.

Depth cue: At the initialization step the proposed technique assumes that the hand is the closest object in the scene to the Kinect sensor, and approximately at the centre of the image. Then for a depth frame D at a time t, it assumes that the hand will be within a cuboid region. The dimensions of the cuboid region are defined as 10 $pixels$ around the X and Y directions, and 5 cm around the $depth/Z$ direction from the hand at previous frame, i.e. hand at time $t-1$. The use of a cuboid mask instead of a spherical mask makes the query of image pixels easier and also the hand is more likely to move either up/down or left/right faster than in diagonal directions. The use of depth information D to create a cuboid mask is known as depth cue in the next sections.

Hand region segmentation: given the skin and the depth cues described above, the proposed technique extracts the largest region which is later provided as an input for the hand pose estimation module (ref. Section 5.3.2).

5.3.2 Hand Pose Estimation

The technique proposed in this chapter uses random forest (Breiman 2001) to regress the hand joints. Random forest is an ensemble of decision trees. Criminisi et al. (2011) published a detailed tutorial on random forest. Following Gall & Lempitsky (2009) and Girshick et al. (2011), the proposed technique uses classification nodes to split a tree, and a Hough voting technique at leaf nodes of the tree for joint proposals. Since using all votes from the training pixels is very difficult to deal with, due to limited memory and available processing power, reservoir sampling (Vitter 1985) has been used. The details of reservoir sampling are described in Section 2.4.2 of Chapter 2. The mean-shift mode finding technique proposed in Comaniciu & Meer (2002) is then applied on those joint proposals. The highest scoring mode of each joint is used for pose estimation, and for the hand tracking, five high scoring modes are used. The details of the mean-shift algorithm is described in Section 2.4.3 of Chapter 2. The feature details, training and testing methodologies are described below.

Depth feature

The quality of features has a significant influence on the quality of hand parts classification. However, because of the computational demand of random forests, simple features are used to achieve real-time computation. Hence, an efficient depth comparison feature from Shotton et al. (2011) is used, which requires only five arithmetic operations. For a pixel d of depth image D, the depth value at d is denoted by $D(d)$, and the depth difference feature is denoted by $\theta = (u, v)$. Here, d represents the 2D location (x, y) of depth image D and similarly u and v are 2D pixel offset values. Then the feature value F is defined as follows:

$$F_\theta(D, d) = D(d + \frac{u}{D(d)}) - D(d + \frac{v}{D(d)}) \tag{5.3}$$

The division by depth value of a given pixel d makes sure the feature is depth invariant. However, since the feature (ref. Equation 5.3) is not rotationally invariant, all possible samples of the targeted application are provided. The maximum length of an adult hand is 23 centimetres (Army 1978), which is approximately 120 pixels at 1 meter distance in Kinect images. Hence, a threshold is applied to F_θ such as $-25cm \leq F_\theta \leq +25cm$ and window size for u and v is 120 pixels per meter (i.e. 240 pixels at 0.5 meter and 60 pixels at 2 meters). The values of u and v are uniformly sampled for the given window size.

Classification forest

Each decision tree of the random forest is trained using the depth difference feature described above to classify the 21 hand regions (ref. Figure 5.1 on page 95). Each split node of a decision tree is trained with a collection of *depth features* and thresholds τ, the aim of thresholds τ is to split all training pixel examples to *left* (L) or *right* (R) child nodes in-order to reduce the uncertainty of the hand region classes C. The proposed technique uses Shannon entropy, S, to measure the uncertainty of 21 hand region classes (ref. Figure 5.1 on page 95). It is defined as:

$$S = -\sum_{c \epsilon C} p(c) log(p(c)) \tag{5.4}$$

where, $p(c)$ is a normalized discrete probability of a hand region class, calculated using the histogram of all training examples at the given node i.e corresponding to the training points in S. Hence, $p(c)$ is obtained as follows:

$$p(c) = \frac{\text{Number of points belonging to class c at S}}{\text{Total number of points belonging to S}} \tag{5.5}$$

Hence, the information gain I of the split node is defined as:

$$I = S - \sum_{i \epsilon \{L,R\}} \frac{|S^i|}{|S|} S^i \tag{5.6}$$

Finally, each split node chooses the best combination of a *depth feature* and a threshold τ, which maximizes the information gain.

Regression of joints positions at leaf nodes

Unlike a regular classification tree, which stores the discrete probabilities of all classes (example hand regions in case of this chapter) at the leaf node, the proposed technique stores 3D offsets for each joint (i.e regression of joint position). However, the voting joint position from long distance is not reliable, hence the votes beyond a defined distance threshold are discarded. Different sets of voting thresholds are used for training and testing and are separately defined below. The leaf node training and testing techniques are described below.

Training:

Algorithm 1: Learning regression of joints positions at leaf nodes

```
for all pixels $d$ in training images do
    for all joints $j$ do
        Lookup the ground ground truth $P_j(x, y, z)$ for joint $j$
        Un-project the pixel $d$ to 3D space $P_d(x, y, z)$ using the Kinect camera calibration matrix provided by the Kinect software development kit
        Compute the relative/voting offset $\Delta_{lj} = P_j(x, y, z) - P_d(x, y, z)$ for joint $j$ at leaf node $l$
        Discard the voting offset if it's absolute value is larger than given threshold value
        Store the voting offset $\Delta_{lj}$ for joint $j$ at leaf node $l$
for all leaf nodes $l$ and joints $j$ do
    Cluster voting offsets $\Delta_{lj}$ using mean shift
    Take top K/2 weighted $w_{lj}$ voting offsets
```

The split nodes of a decision tree are learned using the classification forest technique described above and then the voting offsets for each joint in each leaf node are learned separately. The ideal scenario is to use all voting offsets of training pixels for offset learning; however, it is at a practical level difficult due to the computational complexity. That is why *reservoir sampling* (Vitter 1985), with size 400, is used for offset vote collection following Girshick et al. (2011). In the leaf node l the voting offset Δ for the joint j is defined by $\Delta_{lj} = P_j(x, y, z) - P_d(x, y, z)$, where P_j is the ground truth point in the 3D space for joint j, and P_d is the unprojected point of a given depth pixel in 3D space. The voting offsets are then clustered using a mean-shift algorithm. Similarly to Girshick et al. (2011), two voting offsets from the largest clusters are used and the weight w_{lj} is defined using the number of elements in the cluster; w_{lj} is used to weight the mode/output in the *mean-shift* procedure for joint position prediction. The training bandwidth b_t and the voting threshold

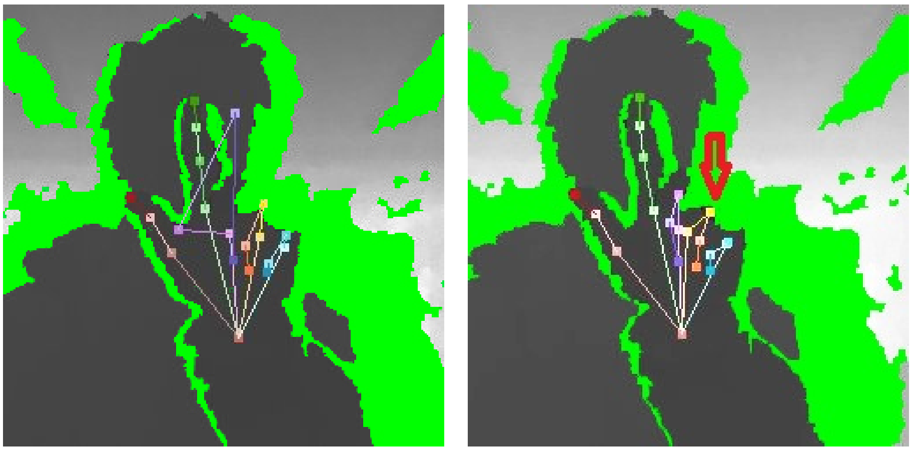
λ_t are learned using a grid search and are the same for all joints.

Joint Inference:

Algorithm 2: Inferences of joints positions

```
for all pixels $d$ in the test image do
    Un-project the pixel $d$ to 3D space $P_d(x, y, z)$ using Kinect camera calibration matrix provided by the Kinect software development kit
    for all trees in random forest do
        Follow the decision tree rule to reach the leaf node $l$
        for all joints $j$ do
            Lookup stored K weighted voting offsets $\Delta_{lj}$
            for all voting offsets K do
                Discard the voting offsets if greater than voting distance threshold
                Otherwise compute proposal joint location $z_j = P_d(x, y, z) + \Delta_{ljk}$
                Adapt weight from training time $Z_j = w_{ijk} * z_j$
// Aggregate weighted votes $Z_j$ for each joint
Sample $Z_j$
Find joint locations using mean shift for each joint $j$
```

In the testing phase, absolute joint proposal points are collected by compensating learned voting offsets from all depth pixel being tested. The weight w_{lj} of the proposal points are re-weighted using the depth value of the pixel as there are fewer pixels for objects further from the sensor. The mean-shift mode finding algorithm is then applied using the highest $N = 500$ weighted joint proposal points which are closer then the test time threshold criteria λ_j for each joint j. The bandwidth b_j and the threshold λ_j for each joint are learned using a grid search.



(a) Using discriminative technique (b) Proposed Technique

Figure 5.2: *This example demonstrates the benefit of combining a discriminative and a descriptive model (MRF).*

Learning parameters

The training time voting distance threshold and mean-shift bandwidth, and test time per-joint voting distance threshold and mean-shift bandwidth parameters are optimized independently. Usually, these parameters are optimized together, but such optimization is computationally expensive. Although this can be seen as a problem, the experiments show that the proposed technique produce good results for hand pose estimation. The grid search is done with cross validation of 2,500 randomly selected hand poses to decide all parameters of proposed technique. Training mean-shift bandwidth b_t = 0.05 cm and the voting threshold λ_t = 15 cm are chosen after grid search with values between 0 to 25 cm. Test time mean-shift bandwidth varies between 0.33 cm to 1.85 cm. Test-time voting thresholds varied significantly, from as low as 1.99 cm to as high as 8.75 cm. The used values of test-time bandwidths and thresholds are as follows (in cm; order: palm joint, thumb metacarpals to little finger tip):

Joint locations	**Bandwidths**	**Thresholds**
Palm	0.68	6.31
Palm centre	0.33	8
Thumb Metacarpophalangeal	0.45	3.78
Thumb Distal Interphalangeal	0.87	2.0
Thumb Tip	1.85	2.04
Index Metacarpophalangeal	0.33	8.75
Index Proximal Interphalangeal	0.73	4.77
Index Distal Interphalangeal	0.35	3.8
Index Tip	0.75	2.96
Middle Metacarpophalangeal	0.33	7.31
Middle Proximal Interphalangeal	0.92	3.08
Middle Distal Interphalangeal	0.8	3.18
Middle Tip	0.92	2.84
Index Metacarpophalangeal	0.33	7
Index Proximal Interphalangeal	0.43	3.6
Index Distal Interphalangeal	0.46	2.8
Index Tip	1.2	1.99
Ring Metacarpophalangeal	0.34	5.85
Ring Proximal Interphalangeal	0.33	3.68
Ring Distal Interphalangeal	1.06	2.43
Ring Tip	0.68	2.49

5.3.3 Hand Tracking

3D hand pose estimation (ref. Section 5.3.2) would be an ideal solution for hand tracking, which can easily overcome the problems of tracking failure and initialization. Unfortunately, due to depth sensor noise and high inter-finger occlusion, pose estimation fails. To improve hand pose estimation, hand-parts kinematics and temporal motion constraints are incorporated. In the initialization phase, the proposed technique expects the hand to be approximately in the center of the frame, after the initialization hand part lengths are estimated/calculated for the next 30 frames. Later, those hand part lengths are used as hand-parts length prior in consecutive frames. The MRF module is then applied, which incorporates joint proposals for the pose estimation module (ref. Section 5.3.2), hand-parts prior and temporal constraints as described below,

Temporal Coherence: Two additional joint proposals j_{t-1} (last position) and $j_{t-1} + v_j$ (projected position), where v_j is the velocity of the joint j and estimated using the joint position in two previous frames, with $s_{t-1} * R_l$ and $s_{t-1} * R_p$ scores respectively added for 50% lower mean shift mode scoring joints in joint inferences procedure; $R_l = 0.4$ and $R_p = 0.5$ are the weights of last position and projected position scores respectively. Experimentally it has been found that assigning higher weight to the projected position than to the previous position produces better results. Besides, increasing last and projected position weights provide some stability against noise but perform poorly under high occlusion and when hand/joint changes direction. Some parts of the Kinect depth image are corrupted by noise, hence optimizing the hand-pose estimation only by using joints proposals from pose estimation, does not produce smooth results, whereas addition of the temporal coherence feature improves the result.

Joint Potential: The joint/unary potential ϕ is defined as:

$$\phi_i(u_i) = \frac{1}{1 + e^{-x}} \tag{5.7}$$

where x is $\frac{s}{\sigma_s}$. Further, s is the score of the joint position hypothesis and $\sigma_s = 0.015cm$ is the score noise.

Kinematic Constraints: The structural connection between hand joints i and j are modelled as *kinematic constraints*, which are defined as:

$$\psi_{i,j}(u_i, u_j) = e^{-(\frac{diff}{\sigma_{diff}})} \tag{5.8}$$

where $diff$ is the difference between hand-part length estimated at hand initialization step, and current prediction (ref. MRF model Figure 5.1 on page 95). $\sigma_{diff} = 10cm$ is the noise of a hand-part length estimation. The value of σ_{diff} is searched between 0 to 20 cm using grid search.

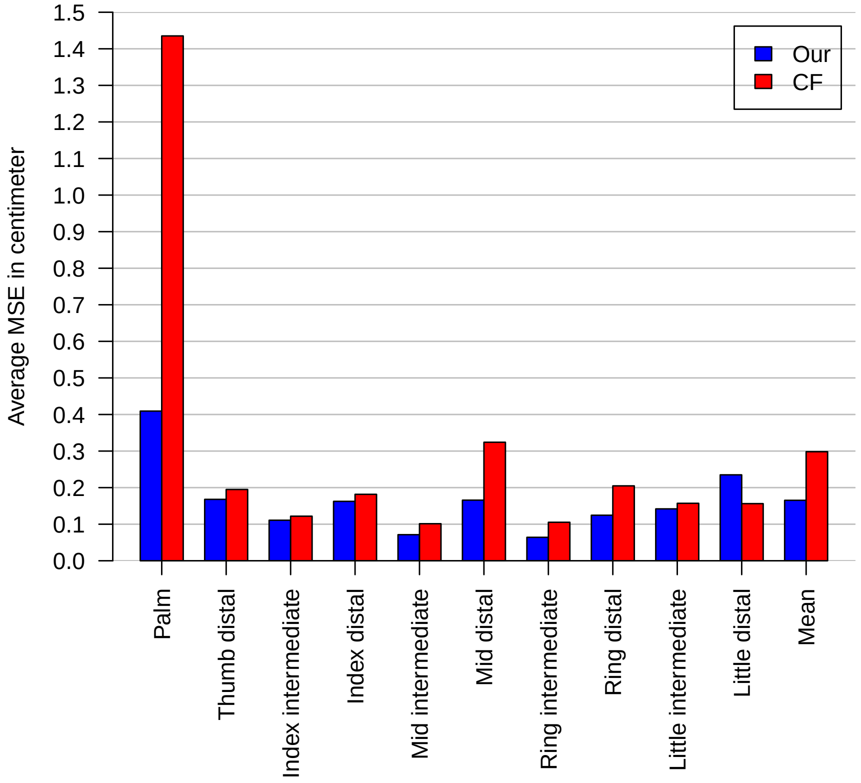
Message passing algorithm i.e. *belief propagation* (Yedidia et al. 2005; Mooij 2010) has been used to maximize the Equations (5.7) and (5.8) to predict the joint positions. The message passing algorithm is briefly described below and detailed in Yedidia et al. (2005). A joint i with $N(i)$ neighbours, sends a message to a neighbour $j\varepsilon N(i)$ when it gets messages from all nodes except j. The message from i to j, $m_{i\to j}(u_j)$, for a joint proposal u_j is defined as:

$$m_{i\to j}(u_j) = \sum \phi_i(u_i).\psi_{i,j}(u_i, u_j) \prod_{k\varepsilon N(i)\backslash j} m_{k\to i}(u_i) \tag{5.9}$$

Finally, the belief of a joint position is obtained as follows:

$$b_i(u_i) = \phi_i(u_i) \prod_{j\varepsilon N(i)} m_{j\to i}(u_i) \tag{5.10}$$

The proposed technique uses only one maximum scoring joint position for 50% higher scoring joints (fixed nodes), and the five higher scoring joint positions (modes of mean-shift) plus two additional positions as explained in the temporal coherence section for the remaining 50% nodes. This strategy allows the proposed technique to give more weight to the discriminative technique and recover the best possible hand pose using kinematic constraints and temporal coherence. Furthermore, the proposed technique does not use positional constraints for the joints as that would violate the tree-structure of the MRF model, and also increase the processing time of *belief propagation* (Mooij 2010) from 2.5 milliseconds (ms) to 6 ms per frame for our experi-



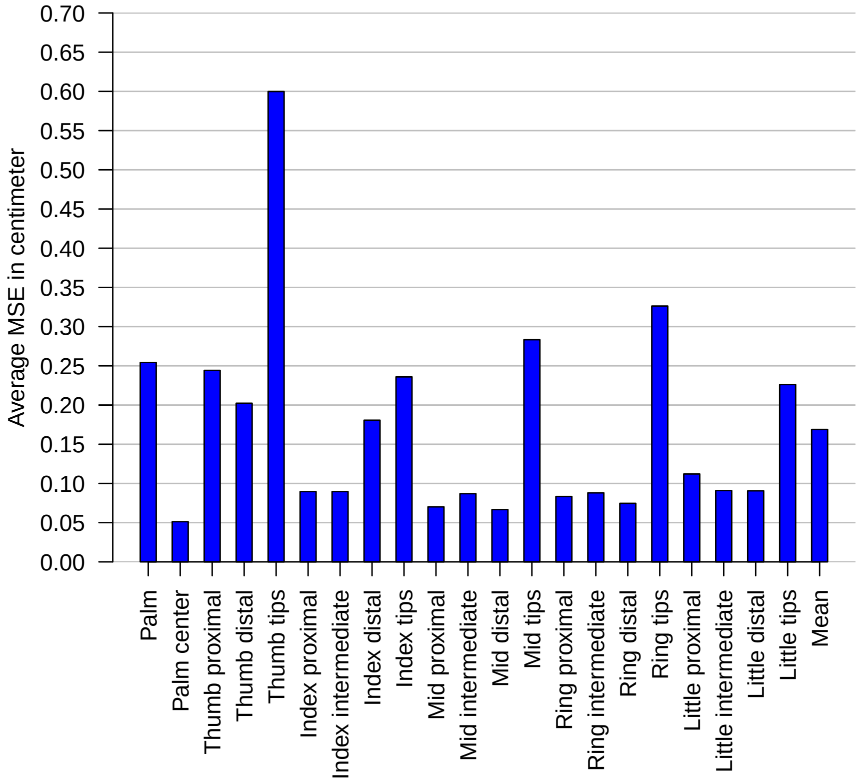
(a) MSEs of our vs CF techniques

Figure 5.3: *Compares the mean square error (MSE) between proposed regression forest and classification forest (CF) techniques for hand pose estimation. Synthetic data has been used for all training and testing (ref. Section 5.3.2).*

ments on a single core 3.33 GHz processor.

5.4 Results and Discussion

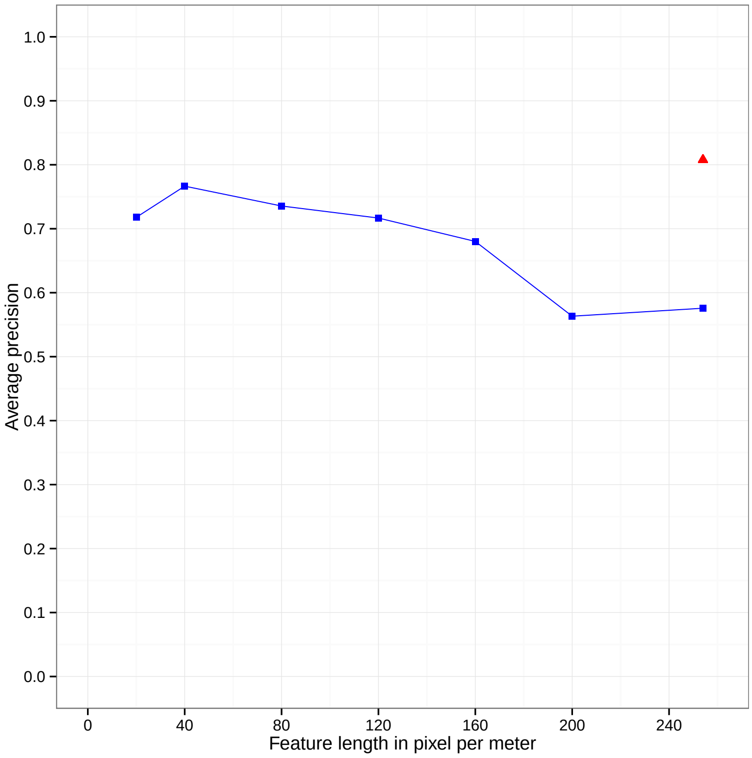
The data for all training are artificially generated as mentioned in Section 5.2. While this can be seen as a drawback, testing in the real world data show that the proposed framework works reasonably well (ref. demo



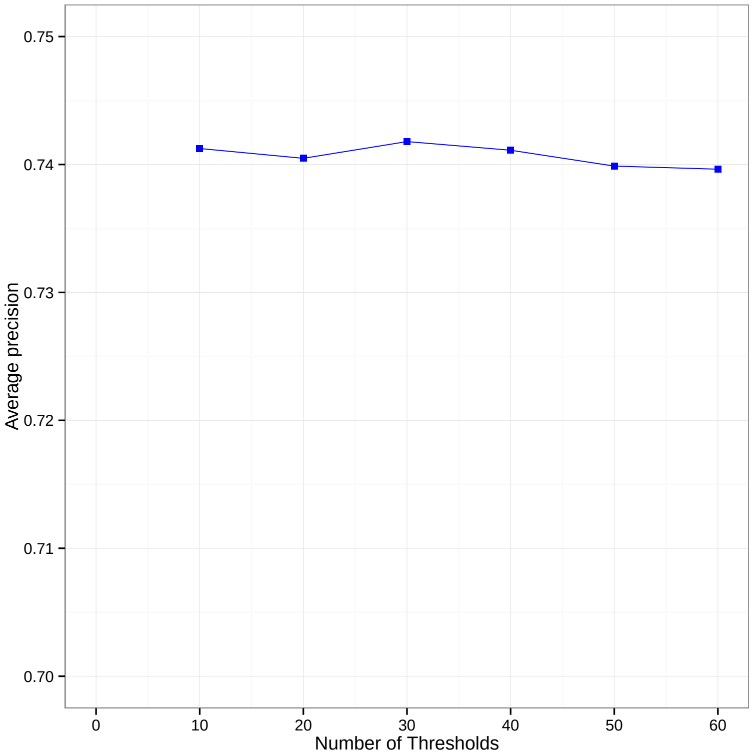
(a) MSEs of our technique

Figure 5.4: *Shows the mean square error of proposed technique for hand pose estimation using 150 thousands data for various hand poses as described in the results and discussions section. Synthetic data has been used for all training and testing (ref. Section 5.3.2).*

http://youtu.be/xqyfWWlAnVI). In this section, first the regression forest and then the classification forest techniques (Keskin et al. 2011) are evaluated to justify the use of the regression forest in the proposed framework. For all experiments, three trees have been used. Even though the addition of more trees increases the performance (Criminisi et al. 2011), the decision to use three trees is based on the computational complexity involved as the training of one tree took nearly 12 days in a largest cluster of Amazon (2014). Also, each regression/Hough-voting tree needed 500 megabyte (MB) of computer memory. Moreover, hand pose estimation system, Keskin et al. (2011), similar to ours, has used the three trees. All trees trained with different poses-spreading, grasping, one finger, two fingers, three fingers, four fingers, pointing with index finger, shooting pose with thumb and index finger in wider directions (rotation angles in degree- along x-axis: -30 back to 85 front; along y-axis: -85 to 85; along z-axis: -85 to 85) in 3D.

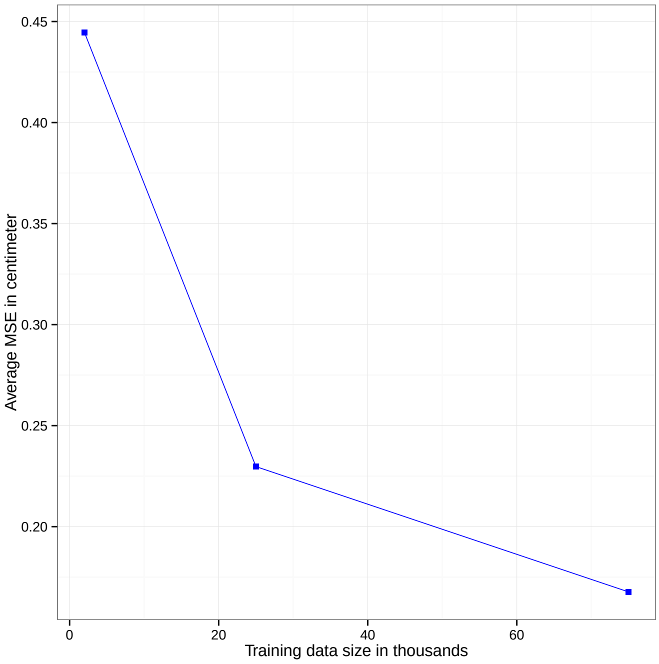


(a) Effect of feature window size

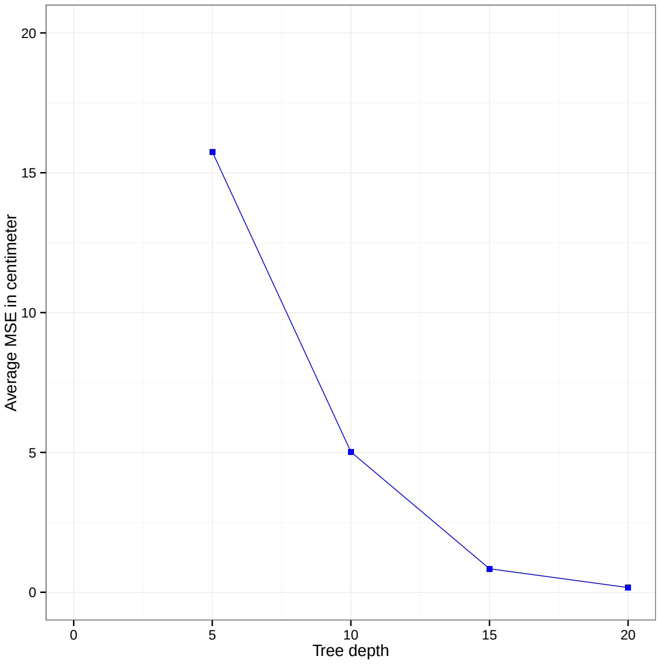


(b) Effect of number of thresholds

Figure 5.5: *Continue to next page*



(a) Effect of training data size



(b) Effect of tree depth

Figure 5.5: *(a) hand-parts pixels classification precision plot against various window sizes. The triangle is a precision of the feature pool technique (use of most frequently used features by split nodes of classification trees from all windows sizes). (b) shows MSEs for the range of thresholds. (c) shows MSE for various training data size. (d) shows MSEs of different tree depths.*

Comparisons with the classification forest based technique: The proposed hand pose estimation technique is compared with the current state-of-the-art hand pose estimation technique of Keskin et al. (2011) for straight hand spreading poses. Three classification trees are trained with tree depth 10 and three thousands synthetic data (ref. Section 5.3.2). As all hand parts are visible, there is no need for employing an occlusion handling module, such as an artificial neural network used by Keskin et al. (2011). Transformation matrices are learned to push joints-prediction from the surface to inside positions. Classification trees are common for both techniques, hence the proposed technique inherits the same advantages and disadvantages created through the above mentioned experimental conditions. The *mean square errors* (MSEs) are presented in Figure 5.3 on page 107. This chapter did not compare the proximal joints because proximal joints are in the middle of the hand-part regions from the back side of the hand, but this is not the same case from the front. This is a more favourable condition to the proposed technique as it uses voting offset rather than finding the center of the defined region as in Keskin et al. (2011). The proposed hand-pose detection technique clearly outperforms the classification forest based technique in estimating the positions of joints except for prediction of the little finger's distal joint. It has been also noticed that if the marked hand-region for training (ref. Figure 5.1 on page 95) is large and the shape is not regular in all directions the MSE is higher.

Feature pool: it has been observed that there is a positive correlation between pixel classification accuracy and the regression of the joint, as the proposed regression forest shares the same classification split nodes. Hence, for the feature selection pixels classification accuracy has been used. Firstly, 3200 features are uniformly sampled with different *window sizes* (i.e. value of feature u and v as described in the Section 5.3.2) and experimented separately; the results are presented in Figure 5.5(a) on page 110. Experimentally, it has been found that even though larger length features are useful, they are more sparse as the number of features is restricted to 3200. Thus the performance decreased. The same number of the most frequently used features are then selected from all experiments with different *window size*, which gave better results. Such a pool of features is used for all other experiments in this chapter.

Unlike other parameters, the number of thresholds τ (ref. *classification forest* sub-section of Section 5.3.2) has very less effect upon the accuracy (ref: Figure 5.5(b) on page 110) of the hand pose estimation. Experiments show that with higher numbers of training images, thresholds between 30-35 work better. In contrast, the tree depth has a significant effect on the accuracy. Due to the computational and memory limitations, the depth of trees is restricted to 20 levels (ref: Figure 5.5(b) on page 111). It has been noticed that with tree depths lower than 10, the classification forest technique performs better than the regression forest based technique. The training dataset size is dependent upon the variation of hand poses as well. The proposed technique works reasonably well when the dataset contains more than 30 thousand training images (ref: Figure 5.5(a) on page 111). Due to the limitation of computational resources, it was not possible to train the proposed framework with more data. It is believed that the accuracy of the proposed framework can be increased with more training data (ref. *effect of data size* Figure 5.5(a) on page 111).

The MSE of the proposed technique is plotted on Figure 5.4 on page 108. Finger tips are likely to be occluded in certain poses more than other hand-parts, hence MSEs of finger tips are higher. Figure 5.6 on page 115 shows a few examples of hand pose regression. These results clearly show how well the proposed technique was able to capture the 3D pose of the hand. Figure 5.2(b) on page 103 shows the benefit of the proposed technique over discriminative techniques. The proposed technique could not recover a good hand pose if the noise continues for more than 4-6 frames, and there are strong false positive joints proposals. Also, the proposed technique fails on hand poses which were not seen during the training time. In the training time the forward movement of a single finger was not provided, hence in the demo (http://youtu.be/xqyfWWlAnVI) it fails in those situations.

5.5 Summary

This chapter presented a markerless 3D hand tracking framework, which efficiently combined discriminative and descriptive techniques. Giving more weight to discriminative technique by fixing high scoring joints/MRF-nodes takes full advantage of the strength of the discriminative technique. Added temporal coherence enables recovery of joints position from noise. Modelling hand joints as unary potential of the MRF model, captures hand-parts length variation efficiently. This chapter also demonstrated that the regression forest based technique outperforms the classification forest based technique for hand pose estimation. To the best of our knowledge, the proposed technique is the first disjoint evidence technique that does not require an additional occlusion handling module for hand pose estimation. It has been demonstrated that the feature pool technique is a simple yet efficient way of generating features from larger feature spaces.

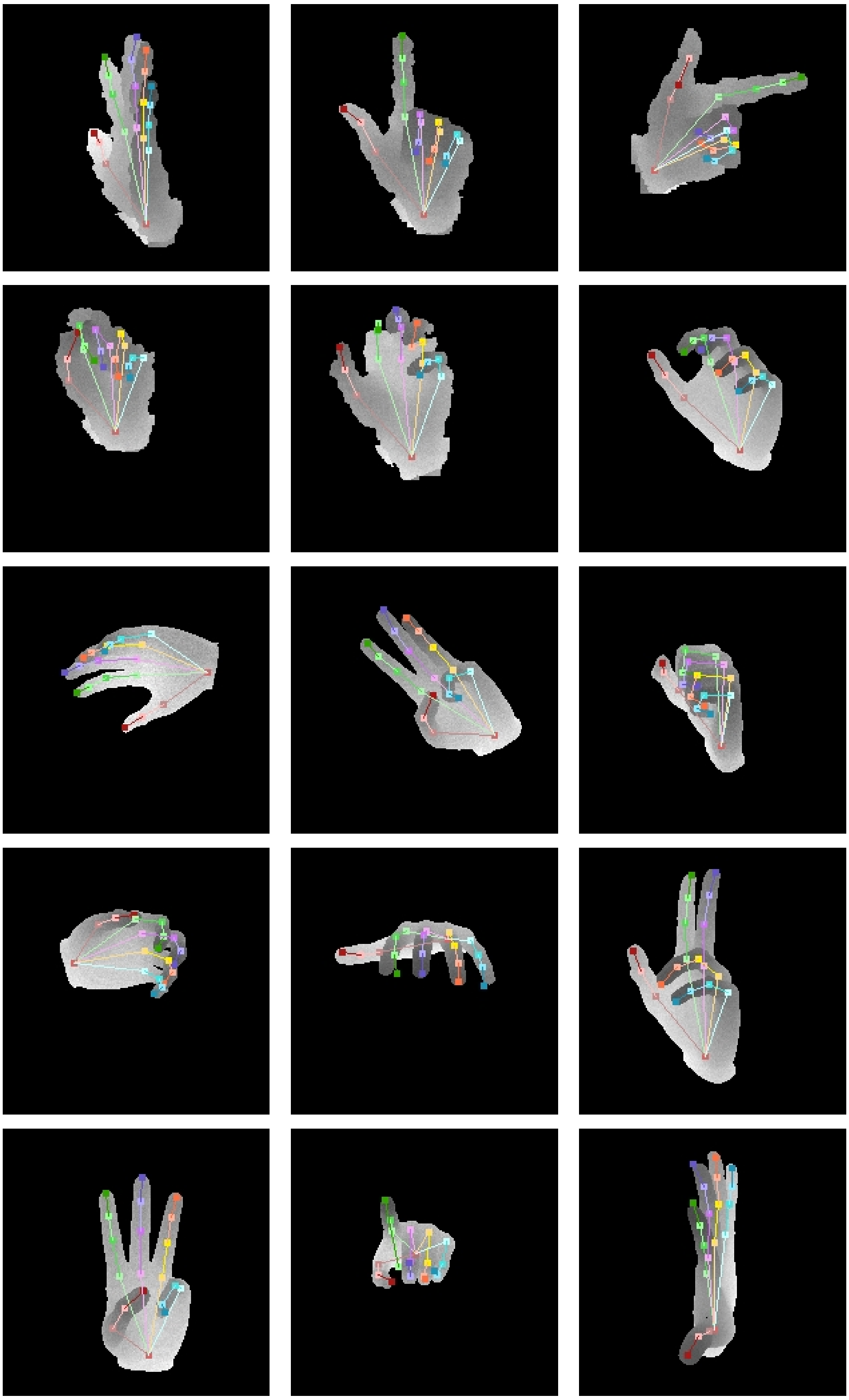


Figure 5.6: *Examples of hand pose estimation. The top row is Kinect depth images and the bottom two rows are artificial data.*

Chapter 6

Conclusion and Future Works

This chapter summarises the work presented in this thesis. The limitations and future works are also discussed.

6.1 Thesis Contribution

This thesis proposed a region-based skin color detection and two markerless hand tracking techniques for human computer interaction. The proposed 3D hand tracking techniques infer the hand joint locations more accurately than existing techniques (Erol et al. 2007; Keskin et al. 2011), which is important for high accuracy demanding applications such as MediKinect (2013).

3D hand tracking is still a challenging problem due to the high inter-finger occlusion, fast random movements and appearance similarity of the hand-parts. The tracking by the detection technique proposed in the Chapter 5 tackles such issues efficiently.

The major contributions of this thesis are: i) a region-based skin color detection technique (ref. Chapter 3); ii) a model-based 3D hand tracking technique (ref. Chapter 4); iii) an appearance based 3D hand tracking technique, which combines the best of the discrimination and descriptive techniques (ref. Chapter 5). These contributions are summarized below.

Skin color provides an important cue for many computer vision applications. Skin color detection is computationally efficient yet invariant to rotation and scaling. The main challenges of skin color detection are illumination, ethnicity background, make-up, hairstyle, eyeglasses, background color, shadows and motion (Kakumanu et al. 2007). Most skin color detection techniques are *pixel-based* and treat each skin or non-skin pixel individually without considering its neighbours. However, skin color is naturally represented as regions instead of individual pixels. This thesis proposed a new skin detection technique based on the concept of regions, irrespective of the underlying geometrical shape. The proposed technique uses a segmentation technique called *superpixels* (Moore et al. 2008; Ren & Malik 2003) to group similar color pixels together. Each superpixel is then classified as skin or non-skin by aggregating pixel-based evidence obtained using a histogram based Bayesian classifier similar to that of Jones & Rehg (2002). However, any suitable skin color classification technique can be used. The result is further improved with *Conditional Random Field* (CRF), which operates over superpixels instead of pixels. Even though the segmentation cost is an overhead over the pixel-based approach, it effectively reduces the processing cost further down the line such as smoothing with CRF. Aggregation of pixels into regions also helps to reduce local redundancy and the probability of merging unrelated pixels (Soatto 2009). Since superpixel preserves the boundary of the objects (Fulkerson et al. 2009), it helps to achieve accurate object segmentation. The presented technique not only outperforms the current state-of-the-art pixel-based skin color detection techniques but also extracts larger skin regions and provides semantically more meaningful results while still keeping the false-positive rate low. This could benefit higher-level vision tasks apart from hand segmentation, such as face and human body detection.

Hand tracking is not a trivial task as it requires tracking of 27 degrees-of-freedom of hand. Hand deformation, self occlusion, appearance similarity and irregular motion are major problems that make 3D hand tracking a very challenging task. Chapter 4 proposed a model-based 3D hand tracking technique. All 16 parts of the hand (one palm and 15 phalanges of five fingers) are sampled and evaluated separately i.e. there are 16 local trackers; such a strategy reduces search space. Each of the hand part samples are evaluated using depth discrepancy features after the hand segmentation. A new depth-fb feature which measures the discrepancy of the background along with the foreground is proposed in Chapter 4. The unexplained regions, segmented hand region/pixels which have not been covered by the predicted 3D hand model, are used to improve the accuracy of hand skeleton prediction. The major contribution of this technique is the use of context cue in the hand tracking. Context cue is used to locate the finger tips and then ICP is used to correct the position of each distal phalanx by keeping the position of other hand parts fixed. This step is called *forward correction*. In the next step, all finger tips/distal-phalanges are kept fixed and other hand parts are searched using the 3D hand tracking technique and it is optimized using Markov random field (MRF). This step is called *backward correction*. Both steps together are named as *forward-backward correction*. Since the shape of the palm is highly deformable, to deal with it, a palm deformation module has been added. The depth-fb feature, context cue and palm deformation module together improved the 3D hand tracking technique but are computationally expensive as the technique requires lots of samples for robust 3D hand tracking. This technique needed 1.8 seconds per frame in a 3.33 GHz Intel processor. To overcome such a problem, appearance based 3D hand tracking technique is proposed in Chapter 5.

Discriminative techniques (for example *random forest*) are good for hand part detection, however they fail due to sensor noise and high inter-finger occlusion. Additionally, these techniques have difficulties in modelling kinematic or temporal constraints. Although model-based descriptive (for example *Markov Random Field*) or generative (for example *Hidden Markov Model*) techniques utilize kinematic and temporal constraints well, they are computationally expensive and hardly recover from tracking failure. Chapter 5 presented a unified framework for 3D hand tracking, utilizing the best of Discriminative and Descriptive techniques. The proposed framework consist of three modules: i) hand region segmentation: segment the hand region using skin and depth cues; ii) hand pose estimation: uses a regression forest to estimate the positions of the hand joints ; iii) hand tracking: uses the pose estimation, kinematic prior and temporal information to track the 3D joints positions. The joint estimation module uses a discriminative *random forest* (Breiman 2001) to classify the hand-parts and learn joints offsets at leaf nodes. *Mean-shift* (Comaniciu & Meer 2002) is used to aggregate the joint votes. The voting distributions are multi-modal in nature; hence, rather than using the highest scoring mode of the voting distribution for each joint separately as Girshick et al. (2011) did, the five high scoring modes of each joint have been fitted on a tree-structure *Markovian model* along with kinematic prior and temporal information. MRF is globally optimized using the approach by Yedidia et al. (2005). Experimentally, it has been observed that relying on a discriminative technique (for example joints detection in case of this thesis) produces better results than generative technique. Therefore, this observation has been efficiently incorporated in the proposed framework by conditioning 50% low scoring joints modes (here it means modes of the *mean-shift*) with the remaining high scoring joints mode. This strategy reduces the computational cost and it can cope with sensor noise, and does not suffer from drifting phenomena. The proposed technique in Chapter 5 does not require additional occlusion or collision handling mechanisms unlike other disjoint evidence techniques of Sudderth et al. (2004a) and Keskin et al. (2011). This technique runs 4-6 frames per second in a 3.33 GHz Intel processor, which can be implemented in multi-core processors or graphical processing units (GUPs) to make it real-time.

The proposed 3D hand tracking techniques in this thesis can be used to extract accurate hand movement features to enable complex human machine interaction such as gaming and virtual object manipulation.

6.2 Limitations

The major limitation of this thesis is that the proposed techniques do not run in real-time in a 3.33 GHz Intel processor. All techniques are implemented in C++ programming language. 3D hand tracking using Markov random field needed 1.8 seconds for one frame. The most time consuming part in this technique is samples rendering and likelihoods calculations of the samples. The 3D hand tracking using random forest and Markov random fields runs in 4 frames per second. The second technique is highly parallelizable. Hence, we believe that the proposed algorithms can easily run in real-time on a multi-core or GPU. Further, the proposed techniques cannot track two hands simultaneously.

6.3 Future Work

This section discusses a number of potential directions for future work. Throughout the experiments involved in the development of this thesis, the following future works have been identified.

Adding texture information for skin color detection: image texture is about the spatial arrangement of color in a selected region of an image. Texture plays an important role in object detection, for an example texture difference between a Giraffe and a Camel. It also plays an important role in providing the context information; for example sky, grass, water and road textures as in Shotton et al. (2006). Moreover, skin regions do not have the same color values; even the skin color pixels within the same superpixel have different color values. In addition, there are many objects which resemble skin color but have very different textures, for example a computer desk. Hence, adding texture information for basic skin color detection or region-based skin color detection is likely to improve the result. However, the improvement in skin color detection accuracy will come at the price of additional computational costs, which in the case of real-time 3D hand tracking might be an issue.

Two hand tracking: mostly humans use two hands for human-to-human interaction. Hence, using two hands for human computer interaction will be a natural choice, in comparison to single hand interaction techniques. At the time of writing, there has been also growing interest towards two hand tracking. Oikonomidis et al. (2012) tracked skeletons of two interacting hands using a template matching technique. It would be interesting to extend the proposed 3D hand tracking technique in Chapter 5 to two hands, and compare this with the template matching technique proposed by Oikonomidis et al. (2012). However, tracking two interacting hands using *appearance based* and *disjoint evidence* techniques (for an example as in Chapter 5) might be more challenging than template matching techniques (for an example see Oikonomidis et al. (2012)), as this is due to occlusion handling being more difficult with appearance-based and *disjoint evidence* techniques, in comparison to template matching and *joint evidence* techniques (Oikonomidis et al. 2012). Hence, in each frame initializing the template using appearance-based techniques (for an example, joint detections using *random forest* as in Chapter 5) then refining the system using the template matching techniques (for an example Oikonomidis et al. (2012)), would be an interesting future direction to pursue since using appearance-based technique would help to overcome *particle degeneracy* phenomenon that occur with the template matching technique, and template matching will complement the occlusion handling module for appearance based technique.

GPU implementation: proposed 3D hand tracking techniques can be implemented in a general-purpose graphics processing unit (GPGPU) to reduce the per frame processing time. The joint detection technique, *random forest* (Breiman 2001), used in this thesis can speed up this up considerably (Girshick et al. 2011). Hence, real-time 3D hand tracking using the proposed technique in Chapter 5 normally could be possible. However, training of the random forest using a larger amount of data does not give much benefit by GPGPU (Sharp 2008). Additionally, MRF optimization in GPGPU might be difficult, hence global hand pose optimization using MRF in Chapter 5 can be replaced by using the template matching technique, i.e. quick hand pose initialization with random forest, and refinement with the template matching technique.

Using multiple depth sensor for hand tracking: one of the major problems of 3D hand tracking is self-occlusion. Many authors have tried articulated hand motion tracking by using multiple RGB cameras (Utsumi & Ohya 1999; Usabiaga et al. 2009; Oikonomidis et al. 2011b) to minimize the element of self-occlusion by the hand. Besides, self-occlusion can be minimized using multiple depth sensors and would be computationally more efficient than multiple RGB cameras (Zhang et al. 2012). Therefore, it would be interesting to consider the effects of multiple depth sensors for 3D hand tracking.

Hand gesture recognition and object manipulation: gesture recognition research has a long history, the summaries of gesture recognition research have been published in Pavlovic et al. (1997), Wu & Huang (1999), Konstantinos G. (2004), Hassanpour et al. (2008), and Garg et al. (2009). Also, the main aim of 3D hand tracking technique is to detect the hand joint positions in 3D space for *human-computer interaction* applications. The applications vary from hand gesture recognition to virtual object manipulation.

Hence, in future, it would be worthwhile to experiment with gesture recognition and virtual object manipulation using the 3D hand tracking techniques proposed in this thesis.

Bibliography

R. Achanta, et al. (2012). 'SLIC superpixels compared to state-of-the-art superpixel methods'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(11):2274 – 2282.

J. K. Aggarwal & Q. Cai (1997). 'Human motion analysis: A review'. In *IEEE Proceedings of Nonrigid and Articulated Motion Workshop*, pp. 90–102. IEEE.

Amazon (2014). 'Amazon Web Services (AWS) - Cloud Computing Services'. http://aws.amazon.com/. [Accessed 16 August 2013].

AMD-GPU (2013). 'AMD GPU Tools'. http://developer.amd.com/tools-and-sdks/. [Accessed 07 July 2013].

U. S. Army (1978). 'Human Engineering Design Data Digest'. *US Army Missile Command, Redstone Arsenal, AL* .

Asus (2013). 'Asus Xtion'. http://event.asus.com/wavi/product/xtion.aspx. [Accessed 07 July 2013].

V. Athitsos & S. Sclaroff (2003). 'Estimating 3D Hand Pose from a Cluttered Image'. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2**:432–9 vol.2, 432.

L. Ballan, et al. (2012). 'Motion Capture of Hands in Action using Discriminative Salient Points'. In *European Conference on Computer Vision*, pp. 640–653.

J. L. Barron, et al. (1994). 'Performance of optical flow techniques'. *International Journal of Computer Vision* **12**(1):43–77.

H. G. Barrow, et al. (1977). 'Parametric correspondence and chamfer matching: Two new techniques for image matching'. In *International Joint Conference on Artificial Intelligence*, pp. 659–663.

H. Bay, et al. (2006). 'Surf: Speeded up robust features'. In *European Conference on Computer Vision*, pp. 404–417. Springer.

J. L. Bentley (1975). 'Multidimensional binary search trees used for associative searching'. *Communications of the ACM* **18**(9):509–517.

L. M. Bergasa, et al. (2000). 'Unsupervised and adaptive Gaussian skin-color model'. *Image and Vision Computing* **18**(12):987–1003.

P. J. Besl & N. D. McKay (1992). 'A method for registration of 3-D shapes'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 239–256.

C. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer, New York.

M. J. Black & A. D. Jepson (1998). 'EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation'. *International Journal of Computer Vision* **26**(1):63–84.

A. Blake (2006). 'Visual Tracking: A Short Research Roadmap'. *Springer* .

R. A. Bolt (1980). *Put-that-there: Voice and gesture at the graphics interface*, vol. 14. ACM.

Y. Boykov & V. Kolmogorov (2004). 'An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9):1124–1137.

Y. Boykov, et al. (2001). 'Fast approximate energy minimization via graph cuts'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1222–1239.

L. Breiman (2001). 'Random forests'. *Machine learning* **45**(1):5–32.

D. Brown, et al. (2001). 'A SOM based approach to skin detection with application in real time systems'. In *British Machine Vision Conference*, vol. 2, pp. 491–500.

J. Cai & A. Goshtasby (1999). 'Detecting human faces in color images'. *Image and Vision Computing* **18**(1):63–75.

G. Caridakis, et al. (2010). 'SOMM: Self organizing Markov map for gesture recognition'. *Pattern Recognition Letters* **31**(1):52–59.

Y. Cheng (1995). 'Mean shift, mode seeking, and clustering'. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **17**(8):790–799.

R. Cipolla & N. J. Hollinghurst (1996). 'Human-robot interface by pointing with uncalibrated stereo vision'. *Image and Vision Computing* **14**(3):171–178.

D. Comaniciu & P. Meer (2002). 'Mean shift: A robust approach toward feature space analysis'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5):603–619.

D. Comaniciu, et al. (2000). 'Real-time tracking of non-rigid objects using mean shift'. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149.

T. F. Cootes, et al. (1995). 'Active shape models-their training and application'. *Computer Vision and Image Understanding* **61**(1):38–59.

A. Criminisi, et al. (2011). 'Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning'. *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114* **5**(6):12.

J. L. Crowley & F. Berard (1997). 'Multi-modal tracking of faces for video communications'. In *IEEE Conference on Computer Vision and Pattern Recognition*.

CUDA (2013). 'CUDA Zone'. http://www.nvidia.com/object/cuda_home_new.html. [Accessed 07 July 2013].

Y. Cui & J. J. Weng (1996). 'Hand sign recognition from intensity image sequences with complex backgrounds'. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 259–264. IEEE.

N. Dalal & B. Triggs (2005). 'Histograms of oriented gradients for human detection'. In *IEEE Computer Society Conference on Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893.

T. Darrell & A. Pentland (1993). 'Space-time gestures'. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 335–340.

L. De Lathauwer, et al. (1994). 'Singular Value Decomposition'. In *Proceedings EUSIPCO-94, Edinburgh, Scotland, UK*, vol. 1, pp. 175–178.

H. Delingette (1994). 'Simplex meshes: a general representation for 3D shape reconstruction'. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 856–859.

J. Deutscher, et al. (2001). 'Automatic partitioning of high dimensional search spaces associated with articulated body motion capture'. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II–669–II–676 vol. 2.

B. Dorner (1994). *Chasing the colour glove: Visual hand tracking*. Ph.D. thesis, Simon Fraser University.

R. Eberhart & J. Kennedy (1995). 'A new optimizer using particle swarm theory'. In *IEEE Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43.

A. A. Efros, et al. (2003). 'Recognizing action at a distance'. In *IEEE International Conference on Computer Vision*, pp. 726–733.

A. Erol, et al. (2007). 'Vision-based hand pose estimation: A review'. *Computer Vision and Image Understanding* **108**(1-2):52–73.

S. S. Fels & G. E. Hinton (1997). 'Glove-talk II-A neural-network interface which maps gestures to parallel formant speech synthesizer controls'. *IEEE Transactions on Neural Networks* **8**(5):977–984.

P. F. Felzenszwalb & D. P. Huttenlocher (2000). 'Efficient matching of pictorial structures'. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 66–73.

Y. Freund (1990). 'Boosting a weak learning algorithm by majority'. In *COLT*, vol. 90, pp. 202–216.

Y. Freund & R. E. Schapire (1995). 'A desicion-theoretic generalization of on-line learning and an application to boosting'. In *Computational learning theory*, pp. 23–37. Springer.

K. Fukunaga & L. Hostetler (1975). 'The estimation of the gradient of a density function, with applications in pattern recognition'. *IEEE Transactions on Information Theory* **21**(1):32–40.

B. Fulkerson, et al. (2009). 'Class segmentation and object localization with superpixel neighborhoods'. In *IEEE International Conference on Computer Vision*, vol. 5.

J. Gall & V. Lempitsky (2009). 'Class-specific Hough forests for object detection'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1022 –1029.

V. Games (2013). 'Virtua Tennis 4 Kinect and Move Hands-On Preview -

GameSpot.com'. http://uk.gamespot.com/virtua-tennis-4/previews/virtua-tennis-4-kinect-and-move-hands-on-preview-6298139/. [Accessed 07 July 2013].

V. Ganapathi, et al. (2010). 'Real time motion capture using a single time-of-flight camera'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 755–762.

P. Garg, et al. (2009). 'Vision based hand gesture recognition'. *World Academy of Science, Engineering and Technology* **49**:972–977.

J. Geng (2011). 'Structured-light 3D surface imaging: a tutorial'. *Advances in Optics and Photonics* **3**(2):128–160.

R. Girshick, et al. (2011). 'Efficient regression of general-activity human poses from depth images'. In *IEEE International Conference on Computer Vision*, pp. 415–422.

Google (2013). 'Google Glass - Home'. http://www.google.com/glass/start/. [Accessed 07 July 2013].

N. Gupta, et al. (2002). 'Developing a gesture-based interface'. *Journal of the Institution of Electronics and Telecommunication Engineers* **48**(3):237–244.

H. Hamer, et al. (2010). 'An object-dependent hand pose prior from sparse training data'. In *IEEE Conference on Computer Vision and Pattern Recognition*.

H. Hamer, et al. (2009). 'Tracking a hand manipulating an object'. In *IEEE International Conference on Computer Vision*, pp. 1475–1482.

L. Hamers, et al. (1989). 'Similarity measures in scientometric research: the Jaccard index versus Salton's cosine formula'. *Information Processing and Management* **25**(3):315–318.

F. Hamze & N. de Freitas (2004). 'From fields to trees'. In *Proceedings of* *the 20th conference on Uncertainty in Artificial Intelligence*, pp. 243–250. AUAI Press.

C. Harris & M. Stephens (1988). 'A combined corner and edge detector.'. In *Alvey Vision Conference*, vol. 15, p. 50. Manchester, UK.

R. Hassanpour, et al. (2008). 'Vision-based hand gesture recognition for human computer interaction: A review'. In *International Conference Interfaces and Human Computer Interaction*, pp. 125–134.

T. Heap & D. Hogg (1996). 'Towards 3D hand tracking using a deformable model'. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 140–146.

O. Hilliges, et al. (2012). 'HoloDesk: direct 3d interactions with a situated see-through display'. In *Proceedings of the ACM annual conference on Human Factors in Computing Systems*, pp. 2421–2430.

D. Hogg (1983). 'Model-based vision: a program to see a walking person'. *Image and vision computing* **1**(1):5–20.

A. Hollister, et al. (1992). 'The axes of rotation of the thumb carpometacarpal joint'. *Journal of Orthopaedic Research* **10**(3):454–460.

R. L. Hsu, et al. (2002). 'Face detection in color images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5):696–706.

Q. Huynh-Thu, et al. (2002). 'Skin-color extraction in images with complex background and varying illumination'. In *IEEE Workshop on Applications of Computer Vision*, pp. 280–285. IEEE.

M. Isard (2003). 'PAMPAS: real-valued graphical models for computer vision'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. I–613–I–620, Madison, WI, USA.

M. Isard & A. Blake (1998a). 'CONDENSATION-Conditional density propagation for visual tracking'. *International Journal of Computer Vision* **29**(1):5–28.

M. Isard & A. Blake (1998b). 'ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework'. In *European Conference on Computer Vision*, pp. 893–908.

M. Isard & A. Blake (1998c). 'A mixed-state condensation tracker with automatic model-switching'. In *IEEE Sixth International Conference on Computer Vision*, pp. 107–112.

B. Jedynak, et al. (2003). 'Maximum entropy models for skin detection'. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 180–193.

R. Johnson, et al. (2011). 'Exploring the potential for touchless interaction in image-guided interventional radiology'. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3323–3332.

M. J. Jones & J. M. Rehg (2002). 'Statistical color models with application to skin detection'. *International Journal of Computer Vision* **46**(1):81–96.

S. J. Julier & J. K. Uhlmann (1997). 'A new extension of the Kalman filter to nonlinear systems'. In *International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, vol. 3, p. 26.

S. J. Julier & J. K. Uhlmann (2005). 'Unscented filtering and nonlinear estimation'. *Proceedings of the IEEE Conference on Machine Learning and Cybernetics* **92**(3):401–422.

P. Kakumanu, et al. (2007). 'A survey of skin-color modeling and detection methods'. *Pattern Recognition* **40**(3):1106–1122.

M. Kass, et al. (1987). 'Snakes: Active contour models'. *International Journal of Computer Vision* **1**(4):321–331.

S. Kawato & J. Ohya (2002). 'Automatic skin-color distribution extraction for face detection and tracking'. In *International Conference on Signal Processing*, vol. 2, pp. 1415–1418.

J. F. Kennedy, et al. (2001). *Swarm intelligence*. Morgan Kaufmann.

C. Keskin, et al. (2011). 'Real time hand pose estimation using depth sensors'. In *IEEE International Conference on Computer Vision Workshops*, pp. 1228–1234.

K. Khoshelham (2011). 'Accuracy analysis of kinect depth data'. In *ISPRS Workshop Laser Scanning*, vol. 38, p. 1.

D. Kim, et al. (2012). 'Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor'. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, pp. 167–176. ACM.

R. Kindermann & J. L. Snell (1980). *Markov random fields and their applications*, vol. 1. American Mathematical Society Providence.

Kinect (2013). 'Xbox Kinect'. http://www.xbox.com/en-US/kinect. [Accessed 10 March 2013].

V. Kolmogorov & R. Zabih (2004). 'What energy functions can be minimized via graph cuts?'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(2):147–159.

D. Konstantinos G. (2004). 'A Review of Vision-Based Hand Gestures'. Tech. rep., York University, Canada.

J. Kramer & L. Leifer (1988). 'The talking glove'. *ACM SIGCAPH Computers and the Physically Handicapped* (39):12–16.

H. Kruppa, et al. (2002). 'Skin Patch Detection in Real-World Images'. In L. Van Gool (ed.), *Pattern Recognition*, vol. 2449 of *Lecture Notes in Computer Science*, pp. 109–116. Springer Berlin / Heidelberg.

F. R. Kschischang, et al. (2001). 'Factor graphs and the sum-product algorithm'. *IEEE Transactions on Information Theory* **47**(2):498–519.

J. Lafferty, et al. (2001). 'Conditional random fields: Probabilistic models for segmenting and labeling sequence data'. In *International Conference on Machine Learning*, pp. 282–289.

K.-T. Lai, et al. (2010). 'Human action recognition using key points displacement'. In *Image and Signal Processing*, pp. 439–447. Springer.

R. Lange & P. Seitz (2001). 'Solid-state time-of-flight range camera'. *IEEE Journal of Quantum Electronics* **37**(3):390–397.

I. Laptev (2005). 'On space-time interest points'. *International Journal of Computer Vision* **64**(2-3):107–123.

B. Leibe, et al. (2008). 'Robust object detection with interleaved categorization and segmentation'. *International Journal of Computer Vision* **77**(1-3):259–289.

M.-Y. Liu, et al. (2011). 'Entropy rate superpixel segmentation'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2097–2104.

D. G. Lowe (1991). 'Fitting parameterized three-dimensional models to images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(5):441–450.

D. G. Lowe (1999). 'Object recognition from local scale-invariant features'. *IEEE International Conference on Computer Vision* **2**:1150.

B. Lucas & T. Kanade (1981). 'An iterative image registration technique with an application to stereo vision'. In *International Joint Conference on Artificial Intelligence*, pp. 679, 674.

J. MacCormick & A. Blake (2000). 'A probabilistic exclusion principle for tracking multiple objects'. *International Journal of Computer Vision* **39**(1):57–71.

J. MacCormick & M. Isard (2000). 'Partitioned sampling, articulated objects, and interface-quality hand tracking'. *Proceedings 6th European Conference on Computer Vision* pp. 3–19.

M. Martin, De La G., et al. (2008). 'Model-based hand tracking with texture, shading and self-occlusions'. In *IEEE Conference on Computer Vision and Pattern Recognition*.

MediKinect (2013). 'Press release, DSCallards Voice and Gesture Recognition in Operating Theatres. Fact or Fiction? | The Faculty of Science, Engineering and Computing'. http://sec.kingston.ac.uk/news/2012/voice-and-gesture-recognition-in-operating-theatres-fact-or-fiction/. [Accessed 23 Aug 2013].

T. B. Moeslund, et al. (2006). 'A survey of advances in vision-based human motion capture and analysis'. *Computer Vision and Image Understanding* **104**(2-3):90–126.

J. M. Mooij (2010). 'libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models'. *Journal of Machine Learning Research* **11**:2169–2173.

A. P. Moore, et al. (2008). 'Superpixel lattices'. In *IEEE Conference on Computer Vision and Pattern Recognition*.

J. Muller & M. Arens (2010). 'Human pose estimation with implicit shape models'. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pp. 9–14. ACM.

C. V. Nguyen, et al. (2012). 'Modeling kinect sensor noise for improved 3d reconstruction and tracking'. In *Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pp. 524–530.

I. Oikonomidis, et al. (2010). 'Markerless and efficient 26-DOF hand pose recovery'. *Asian Conference on Computer Vision* pp. 744–757.

I. Oikonomidis, et al. (2011a). 'Efficient model-based 3D tracking of hand articulations using Kinect'. In *British Machine Vision Conference*.

I. Oikonomidis, et al. (2011b). 'Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints'. In *International Conference on Computer Vision*.

I. Oikonomidis, et al. (2012). 'Tracking the articulated motion of two strongly interacting hands'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1862–1869.

OpenCL (2013). 'OpenCL- The open standard for parallel programming of heterogeneous systems'. http://www.khronos.org/opencl/. [Accessed 07 July 2013].

OpenNI (2012). 'OpenNI'. http://www.openni.org. [Accessed 10 March 2012].

J. O'Rourke & N. I. Badler (1980). 'Model-based image analysis of human motion using constraint propagation'. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (6):522–536.

Z. Pan, et al. (2003). 'Face recognition in hyperspectral images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(12):1552–1560.

V. I. Pavlovic, et al. (1997). 'Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review'. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7):677–695.

PCL (2012). 'Point Cloud Library'. http://pointclouds.org/. [Accessed 14 July 2013].

P. Peer, et al. (2003). 'Human skin colour clustering for face detection'. In *International Conference on Computer as a Tool*.

S. L. Phung, et al. (2002). 'A universal and robust human skin color model using neural networks'. In *International Joint Conference on Neural Networks*, vol. 4, pp. 2844–2849.

R. Poppe (2010). 'A survey on vision-based human action recognition'. *Image and vision computing* **28**(6):976–990.

R. Poudel (2009). 'Real-time hand gesture recognition for small devices'. MSc thesis, University of Sussex, UK.

R. P. K. Poudel, et al. (2013a). 'A Unified Framework for 3D Hand Tracking'. In *9th International Symposium on Visual Computing*, pp. 129–139, Crete, Greece.

R. P. K. Poudel, et al. (2012). 'Region-Based Skin Color Detection'. In *8th International Conference on Computer Vision Theory and Applications*, pp. 301–306, Rome, Italy.

R. P. K. Poudel, et al. (2013b). 'Skin Color Detection Using Region-Based Approach'. *International Journal of Image Processing* **7**(4):385–394.

Primesense (2013). 'PrimeSense Sensor'. http://www.primesense.com/developers/get-your-sensor/. [Accessed 07 July 2013].

L. R. Rabiner (1990). 'A tutorial on hidden Markov models and selected applications in speech recognition'. In *Readings in speech recognition*, pp. 267–296. Morgan Kaufmann Publishers Inc.

D. Ramanan & D. A. Forsyth (2003). 'Finding and tracking people from the bottom up'. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II–467–II–474 vol. 2. IEEE.

A. Ratnaparkhi (1996). 'A maximum entropy model for part-of-speech tagging'. In *Proceedings of the conference on empirical methods in natural language processing*, vol. 1, pp. 133–142.

J. M. Rehg & T. Kanade (1993). 'DigitEyes: Vision-based human hand tracking'. Tech. rep., Technical Report CMU-CS-93-220, Carnegie Mellon University (Pittsburgh, PA).

J. M. Rehg & T. Kanade (1994). 'Visual tracking of high DOF articulated structures: an application to human hand tracking'. In *European Conference on Computer Vision*, pp. 35–46.

J. M. Rehg & T. Kanade (1995). 'Model-based tracking of self-occluding articulated objects'. In *IEEE International Conference on Computer Vision*, pp. 612–617.

X. Ren & J. Malik (2003). 'Learning a classification model for segmentation'. In *IEEE International Conference on Computer Vision*, vol. 1.

J. Romero, et al. (2009). 'Monocular real-time 3D articulated hand pose estimation'. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 87–92.

ROS (2013). 'Kinect Calibration technical'. http://wiki.ros.org/kinect_calibration/technical. [Accessed 07 July 2013].

R. Rosales, et al. (2001). '3D Hand Pose Reconstruction Using Specialized Mappings'. *IEEE International Conference on Computer Vision* **1**:378.

B. Sapp, et al. (2011). 'Parsing human motion with stretchable models'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1281–1288.

N. Sebe, et al. (2004). 'Skin detection: A Bayesian network approach'. In *International Conference on Pattern Recognition*, pp. 903–906, Cambridge, UK.

T. Sharp (2008). ‘Implementing decision trees and forests on a GPU’. In *European Conference on Computer Vision*, pp. 595–608. Springer.

E. Shechtman & M. Irani (2007). ‘Matching local self-similarities across images and videos’. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

J. Shotton, et al. (2011). ‘Real-Time Human Pose Recognition in Parts from Single Depth Images’. In *IEEE Conference on Computer Vision and Pattern Recognition*.

J. Shotton, et al. (2006). ‘Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation’. *European Conference on Computer Vision* pp. 1–15.

L. Sigal, et al. (2003). ‘Attractive people: Assembling loose-limbed models using non-parametric belief propagation’. In *Advances in neural information processing systems*.

S. Soatto (2009). ‘Actionable information in vision’. In *IEEE International Conference on Computer Vision*, vol. 25.

D. A. Socolinsky, et al. (2003). ‘Face recognition with visible and thermal infrared imagery’. *Computer Vision and Image Understanding* **91**(1-2):72–114.

T. Starner, et al. (1998). ‘Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video’. *IEEE Trans. Pattern Analysis and Machine Intelligence* **20**(12):1371–1375.

T. Starner, et al. (1997). ‘A Wearable Computer Based American Sign Language Recognizer’. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, pp. 130–137.

B. Stenger, et al. (2001). ‘Model-based 3D tracking of an articulated hand’. In *IEEE Conference on Computer Vision and Pattern Recognition*.

B. Stenger, et al. (2006). ‘Model-based hand tracking using a hierarchical Bayesian filter’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(9):1372–1384.

H. Stern & B. Efros (2002). ‘Adaptive color space switching for face tracking in multi-colored lighting environments’. In *IEEE International Conference on Automatic Face and Gesture Recognitio*, pp. 249–254.

D. J. Sturman & D. Zeltzer (1994). ‘A survey of glove-based input’. *IEEE Computer Graphics and Applications* **14**(1):30–39.

E. Sudderth, et al. (2004a). ‘Distributed occlusion reasoning for tracking with nonparametric belief propagation’. *Neural Information Processing Systems* **17**:1369–1376.

E. B. Sudderth, et al. (2003). ‘Nonparametric belief propagation’. In *IEEE Conference on Computer Vision and Pattern Recognition*.

E. B. Sudderth, et al. (2004b). ‘Visual hand tracking using nonparametric belief propagation’. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, vol. 12.

M. Sun, et al. (2012). ‘Conditional regression forests for human pose estimation’. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3394–3401.

D.-N. Ta, et al. (2009). ‘SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors’. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2937–2944.

J. C. Terrillon, et al. (2000). ‘Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images’. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, p. 54.

C. Tomasi, et al. (2003). ‘3d tracking= classification+ interpolation’. In *Ninth IEEE International Conference on Computer Vision*, pp. 1441–1448.

J. Usabiaga, et al. (2009). ‘Global hand pose estimation by multiple camera ellipse tracking’. *Machine Vision and Applications* **21**(1):1–15.

A. Utsumi & J. Ohya (1999). ‘Multiple-hand-gesture tracking using multiple cameras’. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1.

A. Vedaldi & B. Fulkerson (2008). ‘VLFeat: An Open and Portable Library of Computer Vision Algorithms’. `http://www.vlfeat.org`. [Accessed 20 July 2014].

A. Vedaldi & S. Soatto (2008). ‘Quick shift and kernel methods for mode seeking’. *European Conference on Computer Vision* pp. 705–718.

J. S. Vitter (1985). ‘Random sampling with a reservoir’. *ACM Transactions on Mathematical Software* **11**(1):37–57.

C.-C. Wang & K.-C. Wang (2008). ‘Hand posture recognition using adaboost with SIFT for human robot interaction’. In S. Lee, I. Suh, & M. Kim (eds.), *Recent Progress in Robotics: Viable Robotic Service to Human*, vol. 370 of *Lecture Notes in Control and Information Sciences*, pp. 317–329.

R. Y. Wang & J. Popovi (2009). ‘Real-time hand-tracking with a color glove’. In *ACM Transactions on Graphics (TOG)*, vol. 28, p. 63. ACM.

X. Wang, et al. (2007). ‘Tracking of deformable human hand in real time as continuous input for gesture-based interaction’. In *International Conference on Intelligent User Interfaces*, pp. 235–242.

Y. Wang & B. Yuan (2001). ‘A novel approach for human face detection from color images under complex background’. *Pattern Recognition* **34**(10):1983–1992.

K. W. Wong, et al. (2003). ‘A robust scheme for live detection of human faces in color images'. *Signal Processing: Image Communication* **18**(2):103–114.

Y. Wu & T. S. Huang (1999). 'Vision-Based Gesture Recognition: A Review'. In *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pp. 103–115. Springer-Verlag.

Y. Wu & T. S. Huang (2000). 'View-independent recognition of hand postures'. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 88–94.

Y. Wu & T. S. Huang (2001). 'Hand modeling analysis and recognition for vision-based human computer interaction'. *IEEE Signal Processing magazine- Special issue on Immersive Interactive Technology* **18**(3):51–60.

Y. Wu, et al. (2005). 'Analyzing and capturing articulated hand motion in image sequences'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(12):1910–1922.

Y. Wu, et al. (2000). 'An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization'. In *Proceedings of Asian Conference on Computer Vision*, pp. 1106–1111.

A. Y. Yang, et al. (2008). 'Unsupervised segmentation of natural images via lossy data compression'. *Computer Vision and Image Understanding* **110**(2):212–225.

M. H. Yang & N. Ahuja (1998). 'Detecting human faces in color images'. In *International Conference on Image Processing, 1998*, vol. 1, pp. 127–130.

M. H. Yang & N. Ahuja (1999). 'Gaussian mixture model for human skin color and its application in image and video databases'. In *Proceedings of SPIE: Storage and Retrieval for Image and Video Databases VII*, vol. 3656, pp. 458–466. Citeseer.

J. S. Yedidia, et al. (2005). ‘Constructing free-energy approximations and generalized belief propagation algorithms’. *IEEE Transactions on Information Theory* **51**(7):2282–2312.

Q. Yuan, et al. (2005). ‘Automatic 2D hand tracking in video sequences’. In *IEEE Workshops on Application of Computer Vision*, vol. 1, pp. 250–256.

M. Zahedi, et al. (2005). ‘Appearance-Based recognition of words in American sign language’. In *Pattern Recognition and Image Analysis*, vol. 3522/2005, pp. 519, 511.

L. Zhang, et al. (2012). ‘Real-time human motion tracking using multiple depth cameras’. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 2389–2395.

Y. Zhang, et al. (2011). ‘Superpixels via pseudo-boolean optimization’. In *IEEE International Conference on Computer Vision*, pp. 1387–1394.

H. Zhou, et al. (2009). ‘Object tracking using SIFT features and mean shift’. *Computer Vision and Image Understanding* **113**(3):345–352.

Q. Zhu, et al. (2004). ‘Adaptive learning of an accurate skin-color model’. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 37–42.

T. G. Zimmerman, et al. (1987). ‘A hand gesture interface device’. In *ACM SIGCHI Bulletin*, vol. 18, pp. 189–192.