Real Time Hand Pose Estimation for Human Computer Interaction

Philip Krejov

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

September 2015

Abstract

The aim of this thesis is to address the challenge of real-time pose estimation of the hand. Specifically this thesis aims to determine the joint positions of a non-augmented hand. This thesis focuses on the use of depth, performing localisation of the parts of the hand for efficient fitting of a kinematic model and consists of four main contributions.

The first contribution presents an approach to Multi-touch(less) tracking, where the objective is to track the fingertips with a high degree of accuracy without sensor contact. Using a graph based approach, the surface of the hand is modelled and extrema of the hand are located. From this, gestures are identified and used for interaction. We briefly discuss one use case for this technology in the context of the Making Sense demonstrator inspired by the film "The Minority Report". This demonstration system allows an operator to quickly summarise and explore complex multi-modal multimedia data. The tracking approach allows for collaborative interactions due to its highly efficient tracking, resolving 4 hands simultaneously in real-time.

The second contribution applies a Randomised Decision Forest (RDF) to the problem of pose estimation and presents a technique to identify regions of the hand, using features that sample depth. The RDF is an ensemble based classifier that is capable of generalising to unseen data and is capable of modelling expansive datasets, learning from over 70,000 pose examples. The approach is also demonstrated in the challenging application of American Sign Language (ASL) fingerspelling recognition.

The third contribution combines a machine learning approach with a model based method to overcome the limitations of either technique in isolation. A RDF provides initial segmentation allowing surface constraints to be derived for a 3D model, which is subsequently fitted to the segmentation. This stage of global optimisation incorporates temporal information and enforces kinematic constraints. Using Rigid Body Dynamics for optimisation, invalid poses due to self-intersection and segmentation noise are resolved.

Accuracy of the approach is limited by the natural variance between users and the use of a generic hand model. The final contribution therefore proposes an approach to refine pose via cascaded linear regression which samples the residual error between the depth and the model. This combination of techniques is demonstrated to provide state of the art accuracy in real time, without the use of a GPU and without the requirement for model initialisation.

Key words: Hand Pose Estimation, Discriminative, Generative, Depth, Combined Decent, Random Decision Forest, 3D Tracking, Hand Segmentation, Pairwise Feature, Occlusion, Multi-touch, Hand Tracking, Sign Language, Model Fitting, 3D, Temporal, Articulated Pose, Kinematics

Email: p.krejov@surrey.ac.uk
WWW: `http://www.krejov.com`

Acknowledgements

I would first like to express my sincerest gratitude to Prof. Richard Bowden for his belief in me and this work. I have been very lucky to have a supervisor with so much motivation, enthusiasm and knowledge. His patience and support are the only reason this thesis was possible.

I must acknowledge the members of CVSSP and the Cognitive Vision Lab at the university of Surrey, your help and assistance knows no bounds. In particular Dr. Andrew Gilbert and Dr. Simon Hadfield who shared their time and insightful knowledge.

I thank my Mother, Father and Uncle for giving me the best opportunities in life and encouraging me in every endeavour. I cannot thank you enough for the direction and encouragement you have given me especially since coming to university. I also thank my far flung family members for their kind support.

To my partner, Tanya, I am indebted to her. She gave me the strength to carry on through late nights and many drafts. She has helped me in many ways to finish this thesis, with both quiet patience and unwavering love.

Contents

Nomenclature

ASL	American Sign Language
ABS	Articulated Body Simulation
AR	Augmented Reality
CRF	Conditional Regression Forest
CDO	Constraint Driven Optimisation
CNN	Convolutional Neural Network
DoF	Degree of Freedom
DSP	Dijkstra's shortest path
GMM	Gaussian Mixture Model
GJK	Gilbert-Johnson-Keerthi distance algorithm
HMM	Hidden Markov Model
HCI	Human Computer Interaction
IR	Infrared
ICP	Iterative Closest Point
KF	Kalman Filter
LED	Light Emitting Diode
PSO	Particle Swarm Optimisation
PCA	Principal Component Analysis
PDF	Probability Density Function
RDF	Randomised Decision Forest
RER	Residual Error Regressor
RBS	Rigid Body Simulation

SLR	Sign Language Recognition
SVM	Support Vector Machine
ToF	Time-of-Flight

List of Figures

List of Tables

Declaration

The work presented in this thesis is also present in the following manuscripts:

[1] Philip Krejov, Andrew Gilbert, and Richard Bowden. Guided Optimisation Through Classification and Regression for Hand Pose Estimation. *In preperation for submittion to IVC*, 2015.

[2] Philip Krejov, Andrew Gilbert, and Richard Bowden. Combining Discriminative and Model Based Approaches for Hand Pose Estimation. In *2015 IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015.

[3] Philip Krejov, Andrew Gilbert, and Richard Bowden. A multitouchless interface: Expanding user interaction. *Computer Graphics and Applications, IEEE*, 34(3):40–48, May 2014.

[4] Philip Krejov and Richard Bowden. Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. In *2013 IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7. IEEE, April 2013.

Chapter 1

Introduction

The human hand is our most versatile tool for manipulating the world that surrounds us. Hand gestures are also a natural part of human interaction. As such, resolving the pose of the hand has been a long-standing challenge for computer vision. Currently, computer vision offers the greatest opportunity for natural, non-intrusive hand pose estimation. Therefore, the focus of this thesis is the automatic estimation of the pose of the hand. In particular, the approaches discussed, aim to localise the hand's parts using depth-based sensors without augmentation, investigating computationally efficient algorithms that allow real-time interaction.

This chapter presents the motivation of the work, objectives of the research and the associated challenges. An overview of the thesis structures is also provided, alongside a breakdown of the contributions of each chapter.

1.1 Motivation

There is an increasing need for new innovative methods of interaction, for example in Virtual and Augmented reality. Both offer the potential for new computer interfaces but methods of interaction are underdeveloped, with hand pose estimation being the next step toward seamless human-machine interaction. The mouse and keyboard are the typical devices used for Human Computer Interaction (HCI), however, they are only suited to conventional computing where users are sat at a desk. Hand pose estimation offers user control without physical contact with the computer and is better suited for collaborative computing. Such technology will provide new applications of computing such as; remote surgery, robotics, home entertainment and communication.

Resolving the hand with minimal computational expense is important for the widespread adoption of hand orientated HCI. Reduced computational cost allows real-time estimation and multiple hands to be tracked. Integration with embedded applications would also be feasible, with gestures replacing remote controls in consumer environments.

There are also important social implications regarding the application of sign language recognition. Distinguishing the pose of multiple hands in real-time has the potential to improve recognition rates, and facilitate gestural communication. Recognition of American Sign Language (ASL) fingerspelling hand shapes is one such application demonstrated in this thesis.

Prior work has drawn comparisons between the challenge of hand pose and body pose estimation. In both cases, the objective is to estimate the configuration of an articulated object with a complex kinematic structure and there are similar requirements for real-time performance and accuracy. While there is overlap in the challenges, there are distinct challenges relating to the hands;

Challenging Variation. The range of possible arm motions results in a large variety of global hand poses. The hand can be observed from a range of global rotations (Figure 1.1(a)) while bodies are typically limited to “feet on the floor” scenarios (Figure 1.1(b)). Thus, extensive datasets (50,000+ images) are required to capture the huge variability of the hand. The hand is also subject to considerably fast motion. Estimation must therefore consider the contrasting cases of rapid hand motion and subtle finger motion.

Complex joint dependencies and range. The hand is comprised of a complex chain of kinematic relationships that can cause large-scale deformation. The hand has many Degree of Freedom (DoF) allowing vastly different poses to be formed. The internal structure of the hand also presents dependencies between joints, which are challenging to model explicitly. The folding of fingers also leads to self occlusion,

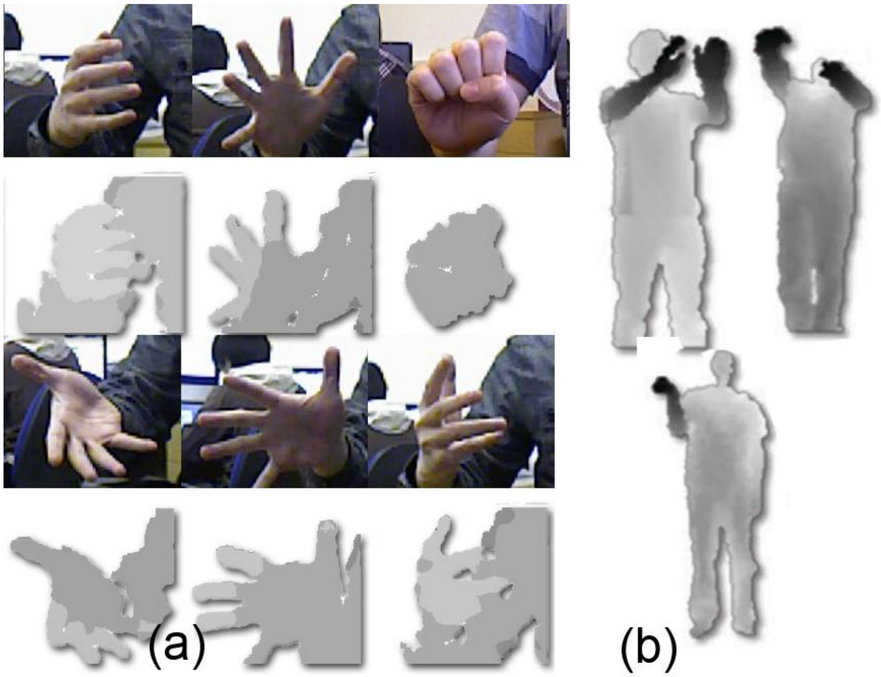


Figure 1.1: (a) Examples of typical hand shapes captured in depth showing noise and large amounts of global rotation. (b) Body poses exhibits less global rotation and better depth estimation.

requiring approaches to have the ability to infer pose for parts of the hand that are occluded. Both dexterity and occlusion lead to ambiguity, which introduces failures in tracking.

An additional challenge using depth sensors is that hands captured using depth can exhibit large amounts of contour noise and missing depth data. This noise is challenging to reproduce in synthetic data [126], and recover when using real data [28]. It is also challenging to obtain large quantities of accurately labelled ground truth data for machine learning approaches.

Lacking Salient Features. With the limited acquisition resolution of depth sensors, and lack of texture, identifying salient parts of the hand is challenging.

Hand pose estimation can however make assumptions that body pose estimation can not, particularly the relatively consistent appearance of hands. Body pose estimation must consider increased degrees of freedom, varied proportions and clothing.

1.2 Depth Acquisition

Using appearance based sensors, many approaches have been proposed to determine both the joints and fingertips of the hand. Such approaches face difficulties due to variance in lighting conditions, and background segmentation. Therefore, a number of them are not suitable for unconstrained environments.

An Infrared (IR) stereo sensor tailored for hand interaction named the Leap Motion [52], provides tracking of fingertips with high accuracy enabling interaction with existing multi-touch applications. However, the Leap Motion has a limited working volume and struggles with complex poses. With the arrival of time-of-flight (TOF) and structured light cameras, depth information has become available and has brought about pioneering methods of capturing user interaction. The depth allows improved background subtraction, facilitating robust isolation of the hand, regardless of background colour. The depth estimate is also invariant to lighting, as the IR light source is actively projected from the sensor itself.

There are several depth sensors now commercially available, each with their own characteristics. The first generation Microsoft Kinect is used in Chapter 1 to provide depth for the Multi-touchless approach. The sensor recovers depth using structured light, which works by projecting a speckle pattern of IR dots. The deformation of this known pattern is then used to recover depth with a resolution of 640x480 pixels. The range of depth that can be recovered is from 500mm to 3000mm. However, due to the typical size of the fingers with respect to the sensor and use of structured light, fingers pointing directly at the Kinect often fail to be recovered. A similar sensor is also available from Asus called the Xtion Motion Sensor.

The second generation Kinect for Xbox One improves the depth resolution and range using time-of-flight, which measures the phase shift of the projected IR light. This allows improved pose recovery at ranges that exceed the previous generation sensor. A close-range sensor provided by Intel called RealSense offers a more suitable range for hand pose estimation. Ranging 0.2m - 1.2m the hand is therefore larger in the depth image providing increased resolution but due to this fact, the device has limited scope for body pose estimation, which can assist hand localisation.

1.3 Contributions

The following section provides a brief overview of each chapter.

Chapter 2 presents a review of the literature concerning hand pose estimation. The chapter draws a comparison between notable works and those discussed in the remainder of the thesis.

Chapter 3 discusses an approach to identify fingertips during rapid movement at varying depths allowing multitouch interaction without contact with a screen. The first contribution is the novel application of a graph based search, to accurately retrieve candidate fingertips. Secondly, we demonstrate an approach to filtering wrist points wrongly identified in the process of selecting fingertip. Its application in a Multi-touchless visualisation system is discussed which allows multimedia data to be browsed using gestures.

Chapter 4 investigates the use of the machine learning for labelling regions of the hand using a RDF. The RDF is an ensemble based classifier which learns features that describe the local and global structure of the hand. The model is capable of learning from thousands of example images while generalising to unseen data. Part based recognition of the hand can then be achieved in real-time allowing its use in initialisation. The model can also store additional information, such as joint offsets or shape labels. The forest is trained using varying sources of data and explores the use of synthetic and glove based automatic labelling. The contributions of the chapter are, the use of a coloured glove as a training aid for labelling vast amounts of ground truth data of the hand, the use of an extended GMM classifier, which integrates noise rejection, and the direct encoding of ASL fingerspelling labels directly in the forest structure.

Chapter 5 presents an approach to hand pose estimation that combines discriminative and model-based methods. This combination aims to remove the need for initialisation and improve model optimisation performance. RDF segmentation forms the basis of fitting a kinematic model to the observation. The models structure is formed of rigid parts, connected using the hand hierarchy. These joints are limited to reflect the limitations imposed by hand kinematics. The first contribution of this chapter is using segmentation to constrain a model solver, which enforces temporal continuity and prevents self-intersection. Secondly, the introduction of cascaded linear regression to improve accuracy, refining part location estimates by removing residual error in model fitting. Finally, intelligent sampling is proposed that represents the models fitting characteristics against various users.

Chapter 6 concludes this thesis, collating and summarising the contributions made in the prior chapters. The chapter also discusses future research, proposing methods for improving the discussed techniques and highlights several long term objectives for the field of hand pose estimation.

Chapter 2

Literature Review

This chapter presents a survey of work relevant to hand pose estimation. There have been a number of methods used in the literature to achieve this. This began with augmentation of the hand with gloves and fiducial markers, allowing the hands parts to be easily distinguished. The demand for unimpeded interaction then led to the use of standard appearance based cameras. Prepossessing such as skin colour segmentation is typically used to capture the shape and contour of the hand. More recently, depth sensors, which where developed for body pose estimation have been explored. The advent of such sensors has improved robustness to varying lighting conditions because depth sensors use infrared light to measure the distance from the sensor. The methods of measurement vary between depth devices, and include; structured light [98], time-of-flight [95] or intensity regression [27]. Many of the following approaches that are applied to depth are inspired by historical appearance based methods.

2.1 Augmentation

In a number of a recent works [42, 95, 114], comparisons have been drawn between the challenge of hand pose and that of body pose estimation. This is because there are similar requirements for real-time performance and accuracy in determining the configuration of an articulated object. However, due to limitations in the resolution and the small area of the fingertips, there are increased challenges in hand estimation as discussed in Chapter 1.

Many of those challenges discussed in the previous chapter can be overcome through augmentation. The addition of salient markers allows robust tracking of the hand, similar to motion capture for body performance. Augmentation can be performed with either active or passive techniques. Active methods apply electronic components to the hand, while passive approaches observe markers using a camera. This thesis will not discuss the use of flex sensors or other mechanical means, and focuses on vision-based approaches.

2.1.1 Active Capture

Using Light Emitting Diodes (LEDs), the position of the hand can be tracked using a video camera. Benko [8] modified a data glove that used flex sensors, with an LED based tracking system allowing the hands complete pose to be recovered. The complete system was rather cumbersome, but demonstrated the validity of hand pose for application control. Inspired by the Minority Report film, Park [77] positioned LEDs on the index, middle finger and palm. The use of multiple LEDs allowed the hand to be localised accurately while gestures could also be recognised. Using LEDs alone greatly reduced the size of the glove. This led to approaches that mount a number of LEDs across the hand. A review of active glove based approaches is presented by Dipietro [23]. With the limited size of the hand, there is an upper limit for the number of LEDs that can be mounted, as placing an LED at each joint is not ideal. Alternatively, using less markers than joints, together with the knowledge of joint limitations, the remaining joints can be resolved. With LED markers placed at the wrist and each fingertip, Aristidou [4] could accurately locate these distinct parts. Then using inverse kinematics, the remaining joints of the hand where estimated.

Another approach to active finger tracking is to place a camera on the user. This improves robustness to global orientation, in turn improving accuracy. Depth cameras can be placed on the shoulder [38] for forward facing interaction and is described as an egocentric mounting. Using relatively simple measurements of the fingers width and contact point with other surfaces, such a mounting allows mobile touch interaction. The camera can also be placed on the wrist [44] for accurate recovery of finger pose. Mounting the camera so close to the hand almost entirely normalises against global rotation. However, mounting such cameras is cumbersome due to their size and because they must protrude from the wrist, so not to be occluded by the palm.

Despite their high level of accuracy and reducing size, active gloves are likely to have an increased production complexity over passive alternatives. This reduces their viability as a solution for general computing. However, they are still popular for motion capture in the film industry and product development [19].

2.1.2 Passive Capture

Due to limited texture information on the hand, it is challenging to discriminate between it and other parts of the body, for example, the face. Therefore, in the past, coloured gloves have been used to help distinguish the hand and improve segmentation. Starner used coloured gloves in his early work on sign recognition [107] as did Bowden and Kadir in [9]. An early approach by Keskin [41] utilised a coloured glove to perform robust tracking of the hand. In many cases subsequent gestures are then recognised using Hidden Markov Models (HMMs).

The use of colour markers has progressed similarly to active pose estimation techniques, being positioned across the hand and reducing in size. Fiducial markers can be placed at key locations on the hand providing features to track. Chua [17] placed a number of coloured markers on the fingertips thumb and wrist. Each marker was a different colour allowing them to be easily distinguished. However, occlusion was not considered, and this lead to failure in tracking. Chua also performed inverse kinematics but with a reduced number of degrees of freedom, to reduce computational complexity. Augmented Reality (AR) markers similar to QR codes have also been used to augment the fingertips allowing pose and rotation to be determined. Piekarski [80] first placed an AR marker on the thumb allowing pinching to be tracked. This was followed by Buchmann [13], who placed a marker on both the index finger and thumb, improving augmented grasping.

As discussed previously there is, a natural limit on the number of markers that can be placed on the hand. Wang [124] proposed the use of a skin-tight glove with a unique pattern. Using a textured model hand, Wang rendered an extensive database of 100,000 hand images. The images themselves were of low resolution, allowing fast comparison. To allow matching against the database, it was encoded using similarity sensitive coding which allowed Hamming distance to be measured. Re-ranking of the 300 best matches was then performed using a Hausdorff-like measure which was more accurate but slower to compute. Poses were then refined with inverse kinematics which incorporated a temporal term. Despite its excellent performance, it is important to note that the bespoke pattern of the glove reduces possible applications. This thesis presents an alternative approach using the glove as a training aid for machine learning and is discussed in Chapter 4.

Limited research has since been conducted with the use of markers with exception of those targeting mobile devices. This is due to its computational simplicity. The recent work of Hurst [40] demonstrates a number of applications focusing on augmented interaction with coloured markers, but gestures are limited, with tracking only being available for the thumb and index finger.

Both approaches proposed by Liang [57] and Tagliasacchi [112] perform augmentation using a wristband to robustly isolate the hand. The wristbands are vivid in colour, ensuring that they are easily distinguished against skin tones. This is not ideal for natural use. However, as show in Chapter 4 it is not necessary as the wrist is differentiable from the rest of the hand. The approach proposed in Chapter 4 isolates the hand using discriminative means, robustly partitioning the hand and forearm.

2.2 Structural Analysis

For unimpeded interaction, approaches were developed using appearance only cameras. As there is limited texture information on the hand, appearance-based approaches tend to rely on information from the edge or contour of the hand to infer structure.

2.2.1 Part Detection

Performing segmentation is challenging when considering complex backgrounds. Therefore robust hand segmentation is required. Using an infrared camera, Oka [75] demonstrated that a strong contour could be captured. This allowed a scanning window to be used to find candidate fingertips and was performed using a template of a rectangle with a semi-circular tip. Oka also demonstrated a method for matching fingertips from one frame to another, using the hand's orientation to find correspondence. Several other bottom-up approaches have also been proposed. Bretzner [12] identified fingertips as blobs using a multi-scale detector. Blobs at multiple scales would be identified as fingertips, fingers and the palm, allowing a hierarchical model of the hand to be formed. The robustness of the model was further improved using a probabilistic skin prior. Many approaches choose to perform skin segmentation as an initial stage [57, 72, 104] . Two widely used skin segmentation techniques are presented by Sigal [100] and Phung [79].

Hierarchical modelling is often performed in body pose estimation. This allows part detections to come together forming a global formation. One notable method was that of Ren [89] that modelled the body using pairwise constraints. These constraints described the spatial relationship between pairs of body parts. Together, multiple constraints model the entirety of the body, and were learnt from training examples. Hackenberg [36] found the fingers and tips using tailored detectors, while Mokhtar [69] used circular templates. Ultimately, the use of part detection and constraint modelling allowed Hamer [37] to estimate occluded hands grasping objects. Using a Markov Random field, pairwise constraints enforced the anatomical structure. Belief propagation was then used to find the optimal pose configuration, with performance decaying gracefully with increased occlusion. Computing these constraints is however computationally intensive, taking several seconds for a single frame. More recent approaches have developed efficient means using hierarchical particle filtering [63] and can operate in real-time without the use of a GPU [82].

2.2.2 Contour and Surface

Pointing is a fundamental gesture that can be robustly detected using both the closest point in a depth image [113], or through Principal Component Analysis (PCA) [39] of the hands silhouette [25, 29]. The fingertips of extended fingers can be found using the contour of the hand, given poses which face the camera. Computing K-curvature, Malik [65] locates peaks and valleys along the contour. The finger orientation is then determined through least-squares fitting of a line against midpoints, found by traversing around the contour. Several similar approaches have been applied to depth images, which improves segmentation robustness [30, 51, 55, 128]. Argyros [3] improved detection and tracking using multiscale curvature, and determined correspondence between frames using an Iterative Closest Point (ICP) derived for contours. Alternatively, ellipses were fitted by Lee [54], providing non-maximal suppression and the improved stability needed for augmented reality applications. Such approaches can also be used for the initialisation of hand tracking [16, 76, 85].

The contour of the hand is subject to acquisition noise for both appearance and depth cameras. This can lead to missed detection of the fingertips. Simion [101] instead computes the midpoint between pairs of contour points mapping the internal area of the fingers. However, the approach only seeks to count the number of fingers shown. A similar approach to finding the middle of the finger is performed by Harrison [38], with an egocentric mounting allowing mobile touch gestures by detecting an unbroken path between fingertips and surfaces. Considering the constrained range of poses that an egocentric camera can observe, relatively simple heuristics can be used to localise the finger. Harrison uses vertical slicing of the depth map's derivative to find edges and label the midpoints of the fingers. However, this approach is dependent on horizontal fingers. Alternatively, convex decomposition was proposed by Qin [86] aiming to perform cuts separating the fingers from the palm. A skeleton could then be extracted using the base of the cut and the fingertip. The skeleton was then used for pose classification of easily distinguished gestures.

With the necessity for a strong contour and non-occluded surface, the previously discussed approaches tend to struggle with out of plane gestures. When using the depth surface, the planar constraints can be more relaxed, allowing fingertip detection for more complex poses. This is because the fingers occluding the hand still present a detectable depth discontinuity [70]. This depth discontinuity was leveraged in body pose estimation approaches that searched for extremities. The first use of geodesic distance was applied to body pose by Plagemann [81]. Using a time of flight camera, the surface of the body was mapped as a graph, with Euclidean edge weightings. Later work by Schwarz [94] and Baak [6] developed improved methods for searching body extrema. Baak proposed the use of a non-initialised distance map, greatly reducing the run-time expense. Schwarz incorporated temporal continuity using optical flow for recovery, which improved robustness further. Both of these approaches also performed skeletal optimisation, explicitly modelling kinematic relationships. Chapter 3 applies this search for geodesic extrema to the task of locating fingertips and extends the approach to provide efficient filtering and was published in [47]. This work in [47] was concurrent with that of Liang [118], which used fingertip detections to initialise a series of particle based trackers.

With surface-based approaches, it is challenging to disambiguate fingers that are next to each other as they form a continuous surface. The approach by Yu [131] aims to alleviate this by manipulating the graphs edge weights, horizontally across the palm. However, this is only suitable for vertical poses. Instead, Liang [59] incorporated appearance information, in which closely neighbouring fingers are separable. The finger locations were then used to initialise a model fitting stage by allowing additional joints to be recovered while imposing kinematic constraints.

2.3 Discriminative Modelling

Many machine learning approaches have been proposed for learning the hands configuration. Discriminative models learn a mapping between observed features and those variables to be predicted. Early discriminative methods performed pose classification, identifying the hand shape against previously seen examples.

2.3.1 Pose Classification

Given an image of the hand, pose classification aims to determine a label that categorises the pose into one of a set of discrete predetermined configurations. For example identifying characters used in fingerspelling. This requires finding unique features which allow a compact representation [49, 129]. An early depth based approach by Malassiotis [64] performed K-nearest neighbour classification against a training set containing 20 alphanumeric signs. Using circular features, a global representation of the hand was captured which was rotationally invariant. This meant that similar poses that differed only due to in plane rotation, could still be matched. The training set used was synthesised using captured poses acquired from a data glove. This allowed an abundance of data to be generated, which is needed to cover the wide pose variation (as discussed in Chapter 1). Using the contour of the hand and Bayesian inference, Guan [35] retrieved similar hand configurations. The approach incorporated the use of a second appearance camera, which dramatically improved performance, highlighting the ambiguity of contour analysis.

Approaches to pose classification have progressed quickly with the introduction of consumer depth cameras. Doliotis [24] proposed a clutter-tolerant hand segmentation algorithm where 3D pose estimation is formulated as a retrieval problem: best matches are extracted from a large database of synthetically generated hand images. Utilising both the appearance and depth, Bergh [122] performed classification by computing Haarlet coefficients which encode the structure of the hand. A discriminative feature projection was then performed, allowing nearest neighbour classification.

Random forests were demonstrated by both Minnen [68] and Pugeault [84] to perform well with both Local and Global features. Inspired by Pugeault, Pedersoli [78] employed Gabor features in combination with Support Vector Machine (SVM) classification to identify the hand shape. This was combined with gesture classification, distinguishing gesture trajectories with a HMM. A review of gesture recognition approaches is presented by [110].

Pose classification can also be used for pose estimation. Matching against a labelled pose allows the joint configuration to be recovered through association. However, is it is difficult to generalise to the range of poses expected during use. Several approaches choose to synthesise their training data as this produces vast training sets with accurate labelling. Proposing two clutter-tolerant indexing methods, Athitsos [5] performed image retrieval from a large dataset. This was only achievable through the use of a low-cost distance measure, which was fast to compute. The matches were then refined using a line based measure that improves discrimination but at an increased cost. An alternative retrieval approach was proposed by Romero [91], using HOG features. Romero also demonstrated that the HOG space lead to ambiguous mapping which was resolved using temporal continuity. This was later improved using a weighted set of prior estimates, allowing the propagation of multiple hypothesis, derived from prior examples [92]. Thippur [119] presents a study comparing Hu moments, HOG and shape feature representations for hand retrieval and discusses their attributes.

Discriminative approaches have since performed region-based detection. This is well suited to the articulated nature of the hand as region-based detection can generalise to other examples. Such approaches can perform joint localisation and implicitly model both kinematic and hierarchical constraints.

2.3.2 Region Classification

These discriminative methods require large amounts of labelled training data, which is often generated synthetically. While synthetic data can provide the extensive training examples required, the quality of the data is heavily dependent on both the physiological accuracy of the model and how closely the data reflects the characteristics of the capture device. To promote realism, Xu [126] incorporates the traits of shadowing and missing depth data, indicative to structured light based depth, while Tang et al [115] explores introducing real data into training using 1.2k manually labelled images. Tang acknowledges that “manually labelled realistic data is extremely costly to obtain” and so combines real and synthetic data using semi supervised learning, which utilises knowledge transfer.

With the recent resurgence of RDFs, real-time localisation and segmentation can be performed. An RDF is a machine learning ensemble, which is comprised of several decision trees [11]. An RDF can be used in a number of ways including classification, regression and manifold learning, which are discussed in the survey by Criminisi [18]. Shotton [98] used depth based features to segment the body into discrete joint based regions. By storing simple features at each tree level, increasingly complex structures could be represented. The span of each decision tree also meant that a large range of poses could be modelled.

Keskin [42] then applied this theory to determine regions of the hand, partitioning it into 21 regions. Using a textured model, Keskin generated synthetic renderings of both the depth and labels. To localise joint positions both Keskin and Shotton performed mean shift ascent for density estimation. Mateusz [66] instead used integral images claiming an improved runtime performance. Localising the joints through mode selection meant kinematic limitations were only modelled implicitly. Keskin also detailed the application of a SVM classifier to identify ASL digits and other easily identified hand shapes. Keskin [43] later extended this by specialising multiple RDFs into cluster based experts. This was an important step in improving the generalisation of random forests. However, the training of multiple forests leads to an increased number of trees, increasing hardware requirements. A alternative solution was presented by Tang [115] which utilised multiple objective functions. Training incorporated clustering into the tree structure, using global pose as the first level objective. As the depth of the tree progressed, the training objective switched to optimising classification and then regression.

The features used in the above approaches all utilise pairwise pixel comparisons of depth, but could also compare depth patches [102], improving their tolerance to camera noise. Yao [127] proposed alternative features that measure surface curvature, which is rotationally invariant. Training was also performed using a coloured glove with segmented regions which separate the fingers and palm. This approach to labelling training data is explored further in Chapter 4.

Region classification is not confined to RDF based methods. The use of a Convolutional Neural Network (CNN) was evaluated by Neverova [71], demonstrating classification with comparable performance. The optimisation of the CNN was driven by two terms, the first, enforced global compactness and homogeneous segments. While the second sought consistency within the local neighbourhood of the predictions. Locally consistent predictions were also sought by Kontschieder [46] and Liang [58], which aimed to reduce classification noise using local geodesic context and a Superpixel-Markov Random Field respectively.

2.3.3 Joint Regression

The location of each of the joints can be regressed directly by discriminative means. Tompson [120] trains a CNN with real labelled data that is annotated using a much slower generative approach. The joints of the hand could also be localised using a regression based RDF, first demonstrated for body pose [34, 99]. This was performed by storing positional offsets from training samples to their respective joints. The multi-objective RDF proposed by Tang [115] optimised over joint variance at the final levels of tree optimisation. This assumes unimodal distributions of offsets at the leaf nodes and is contrary to Girshick's findings, where better accuracy was achieved using region labels as the training objective [34].

Utilising the kinematic structure of the hand improved mode selection. Kirac [45] used bone length to select optimal configurations, as opposed to the globally maximum mode selection, chosen in previous approaches. Poudel [83] performed mode selection using a Markovian model, enforcing kinematic constraints and temporal cohesion. Tang [114] also incorporated hierarchical information. Partitioning the hand using a Latent Tree model, the hands hierarchy could be built into the structure of each regression tree during forest training, such that a single-pass of the forest could recover all of the joints. CNNs can also model such structure implicitly by introducing a bottleneck layer, in practice reducing the variance in the poses to a lower dimensional space. Using both the depth and segmentation, Liang [56] performed a second stage learning, regressing the joints through modelling region correlation. This would predict the joint locations while considering discriminative regions that were already labelled.

These examples of data-driven approaches which incorporate hierarchical structure could potentially fail in generalising to unseen examples, because by their very nature, they are trying to constrain predictions to known configurations. For this reason, the approach discussed in this thesis introduces the combination of discriminative region labelling with generative model optimisation, allowing kinematic, temporal and hierarchical constraints to differentiate ambiguous modes. This is discussed with further detail in Chapter 5

These approaches operated using depth, which limits their use in mobile applications. Song [104] redefined the depth-based features, such that they could operate on skin segmentation masks. This meant depth normalisation had to be discarded, which previously ensured the features were scale invariant. An initial stage classifier was trained to provide feedback to ensure the hand was the correct operational range, where training data modelled the variance in scale. Features could then be sampled for both pose and part classification. Rotational invariance was also incorporated, rotating using the principal axis hand through PCA, but could also be regressed [126]. Liang [57] later presented egocentric appearance-based joint regression using a Conditional Regression Forest (CRF) which incorporated a hidden latent variable, allowing the pose and distance to be inferred jointly.

2.4 Generative Model Optimisation

Generative methods determine the hands configuration using measurements against a model of the hand. The model is commonly rendered with estimated pose parameters typically derived from the previous frames estimate. This operates in an optimisation framework in an attempt to establish the optimal model parameters through refinement. There are a number of benefits in using model optimisation, which include; explicit modelling of the hands kinematics and temporal context. Such processes also allow interaction with objects.

2.4.1 Particle Optimisation

An early optimisation approach was proposed by Rehg [88] which modelled the hand using rigid parts. These parts were connected with joints to represent the hierarchy of the hand skeleton. Each finger was modelled with limited DoF, ensuring the model could only produce valid hand shapes (additional constraints have been developed since [53]). Tracking assumed a limited change in pose which allowed a linear prediction regarding the new position of parts. On acquisition, features were sampled and a residual error vector was determined. A state update was then found through optimisation which reduced this residual error with respect to the kinematic constraints. The approach was later extended to handle self occlusion [87]. Stenger [108] also performed state updates but allowed larger changes in pose between frames by predicting the next pose using an Unscented Kalman Filter, facilitating non-linear prediction. The state of each part could also be updated independently with impact on other joints being deduced through non-parametric belief propagation [111]. However, this was computationally intensive, taking approximately 4 minutes per frame.

As the features in these methods sampled changes in brightness, many required strong background contrast, with tracking performed in front of a black backdrop. A probabilistic framework proposed by Stenger [109] aimed to relax background constraints, and would re-initialise after tracking failure. A tree structure was used to carve the search space so to avoid unlikely poses providing efficient evaluation. To remove ambiguity due to edge contour information, DeLaGorce [20] modelled the lighting and texture of the hand in a constrained fashion, and was shown to improve accuracy. A comprehensive review regarding appearance-based approaches was presented by Erol [26].

It is common for model-based approaches to become trapped in local minima due to the similar appearance of fingers. To recover from such failures Bray [10] proposed the use of stochastic gradient descent computing the response from multiple hypotheses. The optimisation would adjust the step size during updates, to speed up convergence. This was also one of the first approaches to optimise against depth, dramatically improving robustness against background clutter.

Subsequent approaches optimised using Particle Swarm Optimisation (PSO) which provided stochastic descent, in an attempt to avoid local minima. Each particle has a position in the parameter space which is used to evaluate the objective function's score. Oikonomidis [72] constructed an energy term which compared the depth of observed skin coloured pixels (or hand augmented with a coloured glove [90]), against a rendered model. The model was rendered using a GPU allowing many evaluation to be performed for each frame and operated at 15 frames per second. The approach could also be performed in a parametric space of 56 DoF allowing two strongly interacting hands to be estimated. Again, this optimisation was performed on a GPU and performed at 4 frames per second. To improve frame rate, Qian [85] reduced the generative process, using spheres to approximate the shape of the hand. This was combined with extensive sub-sampling of the observed depth, which reduced the cost of calculating the error between the observation and model.

Oikonomidis [74] introduced the use of the Sobol sequence to perform quasi-random sampling. This offered improved characteristics for reduced sample sizes when compared against a uniform distribution, which had been used in previous approaches. In turn, a dramatic improvement in performance was observed when searching high-dimensional parameter spaces.

With increasing demand from the field of robotics, the action of a hand grasping an object has become an important area of research. However, grasping leads to increased difficulties regarding occlusion. The advances in PSO have also allowed estimation of hands grasping simple [73], complex [103] and multiple objects [50]. By considering the constraints between the hand and model of the object, optimisation can be performed as a single system resolving the hand and object jointly. In doing so, knowledge of the hands pose provides information regarding the object's pose and vice versa. This demonstrated that the apparently increased complexity, actually reduced the computational expense and has also been performed with appearance based cameras [125], but again, requires a contrasting background. Due to the features used in discriminative approaches, object interaction leads to degradation of joint localisation.

To improve the optimisation efficiency and realism, Lin [61] proposed reducing the search space using PCA. Learning the variance in the data allows inter and intra finger constraints to be modelled. Dynamic constraints were also modelled, which are difficult to define explicitly with regards to a kinematic model. One such example is when forming a fist, it is more likely that the fingers will fold together. This has since been incorporated into optimisation frameworks to enforce realistic pose transitions [112].

2.4.2 Iterative Closest Point

One of the first approaches to performing 3D articulated Iterative Closest Point (ICP) for the hand was that of Delamarre [22] that presented optimisation of a single finger. Using force to drive model position, the finger model was pulled towards a 3D observation. This is less computationally intensive than particle optimisation approaches, where state updates are determined indirectly from a high-dimensional space. The model was later extended to encompass the complete hand [21], allowing all the joints to be recovered. Loss of tracking was not discussed but must be considered for robust temporaly dependent approaches.

Ganapathi [32] performed ICP for solving body pose estimation. Along with the conventional Physical model of the body, pulled by constraints, Ganapathi proposed a measure of fit, introducing the principle of free space, in which, rays cast from the camera must not occlude background observations. This constraint was later used by Melax [67], alongside several other heuristics to guide state exploration for the hand. Several hypothesis of the pose would be generated simultaneously, each with different kinematic properties. The simulation that had minimal error against the observation was then chosen as the candidate pose. The simulations were constrained using tailored heuristics designed for typical use cases. This however reduced flexibility and led to the model becoming trapped in local minima. ICP surface constraints could be posed as a nonlinear optimisation using the Signed Distance Function [93]. This, however, does not model physical contact between bodies which must be resolved by an additional energy term.

It has been demonstrated that ICP can converge quickly. However, by its very nature, it is highly likely to become trapped in local minima. Using a hybrid approach Qian [85] combines ICP with PSO to improve exploration of the search space. The approach also highlights that ”re-initialisation on every frame is critical for robust tracking.” and proposes an extrema based fingertip detection method. These fingertips are then used to initialisation their model parameters with a reduced DoF, ensuring fast detection.

2.5 Hybrid Methods

Several approaches combine both discriminative and generative techniques, to provide smooth and robust tracking. An early approach performing combined optimisation was proposed by Shimada [96] where pose configuration was initialised via pose retrieval and refined using optimisation. Due to the computational expense and limited resources available, such an approach required multiple computers. Subsequent approaches have reduced the complexity, by optimising over partial detection of hand regions.

One such approach by Ballan [7] performed model fitting against edges, optical flow and salient point correspondences. Fingernails are one such example of a salient feature and were localised using a regression forest (Hough Forest [31]). This was then improved by Tzionas [121] allowing two hands to interact. Sridhar instead localised the fingertips using an SVM classifier [106] which was considerably faster, performing at 10 frames per second. The approach was also found to implicitly reinitialize when fingertips were visible, compared to the simulated annealing, required by Ballan.

The computational performance of randomised decision forests has allowed rapid segmentation of the hands regions. Subsequently, several approaches perform generative optimisation and validation against the forest result. Xu [126] verifies candidate poses through minimization of an energy term derived from the observed and synthesised depth. However, their approach to minimization is computationally expensive allowing only 20 candidates to be minimised at 12 frames per second.

Work conducted in human pose estimation also provides insight for tracking of hands. Taylor [116] regresses continuous labels that correspond to unique positions on the body. This provides dense correspondence allowing the optimisation of a model stood in a default pose with outstretched arms. This research intentionally omits temporal continuity, discarding prior estimates, which offers invaluable information. Chapter 5 presents an approach to efficient minimization of forest segmentation using ICP optimisation. This has since been followed by several approaches minimising labelled hand regions [105]. An alternative approach that offer similar benefits is presented by Sharp [95]. The technique retrieves candidate poses which are minimised using an occlusion aware depth measure. Optimisation is performed using PSO in which particles from prior estimates are carried over, to incorporate temporal information. However, this method of minimisation requires a GPU in order to generate the thousands of particle hypotheses.

2.6 Conclusion

To summarise, hand pose estimation has been conducted using a number of techniques. Early approaches utilised augmentation of the hand. Those that chose active markers were able to demonstrate the viability of gesture control for applications, tracking with a high degree of accuracy. However, these approaches were cumbersome and expensive, detracting from their use in mainstream computing. This led to the development of passive augmentation, providing a low-cost alternative, often using web cameras as the capture device. To achieve natural interaction, no augmentation was favoured. This proved challenging due to complex backgrounds and varying lighting. Utilising skin segmentation and depth based sensors, researchers were able to analyse the shape and contour of the hand, in order for part based models to be fitted. Around this time, approaches sought to recognise poses against a limited vocabulary of known examples. This was often posed as a dataset retrieval problem, utilising discriminative methods. This later extended to continuous pose spaces but required extensive datasets, which were often synthetically rendered. Modelling such large datasets was computationally intensive, requiring approximate measures to allow real-time retrieval. Generative approaches on the other hand, required no training data. Such approaches are able to exploit kinematic and temporal information to improve computational efficiency. However, minimisation techniques can become trapped in local minima due to their dependence on said information. This often required manual reinitialization, as such failures were difficult to detect. The most recent works have opted for a combined approach, exploiting the benefits of multiple techniques. This reduces the mutual failure cases, leading to more robust frameworks.

Chapter 3

Multi-touchless Finger Tracking

Multitouch technology is widespread across modern computers, providing direct interaction with applications spanning mobile and desktop computing. However, interaction is limited to the two-dimensional plane of that display. With the widening spread of computers there is a need for tracking of fingertips independent of the display. Finger tracking in three dimensions opens up multi-touch to new application areas. Removing the need to physically touch the screen would allow new modes of computing ranging from medical analysis in sterile environments to home entertainment and gaming. There is also potential for interactive spaces, with contactless installations and Virtual Reality. The additional depth dimension also provides new avenues for user interface design.

The proposed approach detects only fingertips, but it does so without the use of machine learning. Operating through an entirely structural method, no training data is required. The Multi-touchless approach employs a robust, real-time methodology using depth information. The tips of extended fingers are considered geodesic extrema of the surface of the hand. Using a graph based approach, the extrema can be found and filtered. Tracking fingertips using geodesic maxima instead of their visual appearance is both efficient to compute and robust to both varying poses and contour noise which is typical of depth sensors.

This method of tracking enables interaction with surfaces that are beyond the physical reach of the operator. Providing greater scope for natural interaction as the working space of the user can easily be extended to walls or the entire environment around the users.

This chapter presents three main contributions: Firstly, a novel application of Dijkstra's Shortest path to find candidate fingertips. Secondly, an approach for filtering wrist points that are wrongly identified as extrema. And finally the integration and development of a system for exploring multimedia data using Multi-touchless interaction. The approach operates at 30 fps. Running on a consumer desktop PC, it is capable of processing four hands simultaneously. The techniques developed are designed to be efficient to compute while having the ability to generalise to a range of poses.

3.1 Method Overview

The following section provides an overview of the Multi-touchless approach. The approach analyses images of the hand captured using a depth sensor to find geodesic extreme that correspond to the fingertips. The hand and wrist are segmented from the background, by estimating the users working volume. The depth for a segmented hand can be seen in figure 3.1(a) which includes the hand and wrist. This working volume is established using facial tracking. The hands surface is represented using a connected graph of the three-dimensional points, with edge connections encoding the neighbourhood connectivity (figure 3.1(b)). An efficient derivative of Dijkstra's shortest path algorithm is used to perform a search over the surface of the hand for multiple geodesic extrema. Additional extrema are included to account for extrema that reside around the wrist, and can be seen in figure 3.1(c). The falsely identified fingertips are rejected using highly efficient filtering. The approach considers the path through an elliptical model shown in figure 3.1(d). The resulting fingertip locations (figure 3.1(e)) are then used to update a series of Kalman filters that enforce temporal continuity and models noise characteristics. Frame to frame correspondence is determined using the filter prediction, allowing consistent tracking, necessary for interaction. The tracking is then projected to the image plane for visualisation and integration with existing two-dimensional multi-touch applications. The path followed by each fingertip over the previous ten frames can be seen in figure 3.1(f).

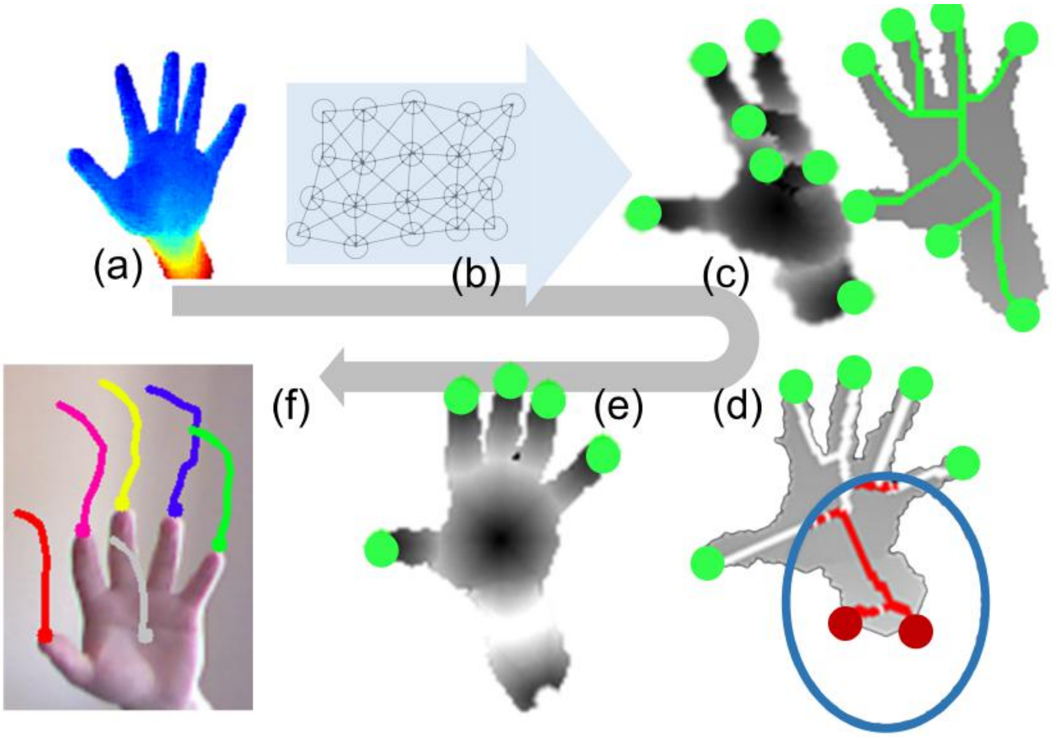


Figure 3.1: Method overview of the Multi-touchless approach showing each stage of the method. (a) Depth is captured using a Kinect sensor and the background is segmented leaving only the hand and wrist. (b) Using a neighbourhood connectivity, each depth pixel is connected to form a graph of the hand surface. (c) Dijkstra's shortest path algorithm searches for a number of extrema. (d) The extrema are then filtered to remove non-fingertips using structural information. (E,F) The final fingertip locations are tracked using a bank of Kalman filters.

3.2 Hand and Forearm Segmentation

The user is captured using a Kinect sensor which uses structured light to acquire a depth image. The value of each pixel in the depth image corresponds to that pixel's distance from the sensors imaging plane. This allows robust background segmentation to be performed, as objects behind a specified distance can be removed. The hands are to be isolated with the depth, by establishing a working volume in front of the user. The users position in front of the camera can be found using standard face detection methods. For each colour frame provided by the Kinect, the user's face is located using a Viola Jones detector [123]. The face detection provides a bounding box of the face, with multiple face detections removed using non-maximal suppression. Using the depth image, the face closest to the sensor $\mathbf{f}$ is found and is considered the working user. The depth of the working user is smoothed over a small temporal window, which reduces noise present in the Kinect's estimate of depth. This depth defines a backplane used for segmentation, allowing the body and background to be removed from the depth image. The remaining depth space in front of this is considered the working area for interaction. Depth images are captured which correspond to the colour image using camera calibration. For the purpose of formulation each pixel in this depth image is defined as $\mathbf{p}$, containing the pixel and corresponding real world coordinates. Using the notation $w(\mathbf{p}) = (x^w, y^w, z^w)$, to represent world coordinates of a point, and $c(\mathbf{p}) = (x^c, y^c)$ for image coordinates. In both cases the use of the subscript $w_i(\mathbf{p})$, $i \in \{x, y, z\}$ provides short hand for a specific dimension.

Points within the working space are connected using connected component analysis. Each connected component is then assigned to a hand candidate which undergoes further analysis. The points of each hand are represented by the set $\mathcal{P}$, hence a hand can be described as the following set of points,

$$\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{|\mathcal{P}|} \tag{3.1}$$

Those components that are smaller than potential hand shapes are rejected with normalisation using $w_z(\mathbf{f})$ which accounts for the user's distance from the Kinect. The components that remain are in practice considered the hand and wrist. Points in $\mathcal{P}$ are then classified as belonging to either a hand or wrist subset defined as $\mathcal{P}^H \subset \mathcal{P}$ and $\mathcal{P}^W = \mathcal{P} \setminus \mathcal{P}^H$. The classification of these points is performed using a depth cut-off W_{depth} positioned at one-quarter of the arms total depth which is calculated using the following:

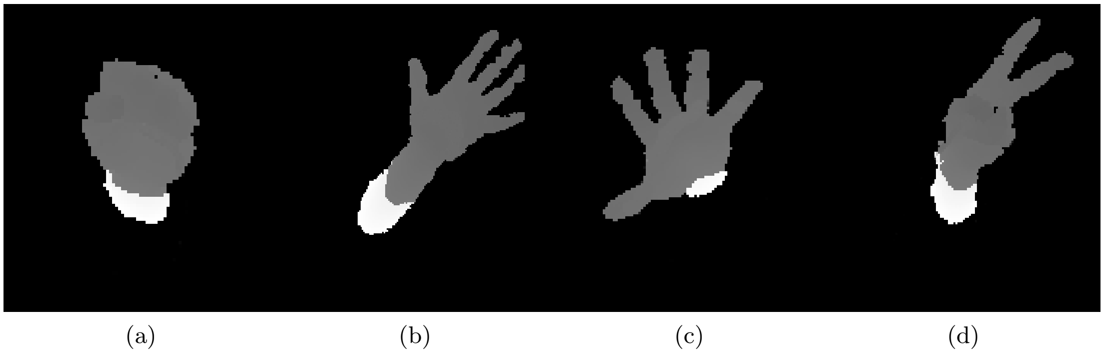
$$W_{depth} = Z_{max} - \frac{(Z_{max} - Z_{min})}{4} \tag{3.2}$$

where,

$$Z_{max} = \max_{\mathbf{p} \in \mathcal{P}}(w_z(\mathbf{p})) \tag{3.3}$$

and,

$$Z_{min} = \min_{\mathbf{p} \in \mathcal{P}}(w_z(\mathbf{p})) \tag{3.4}$$



(a) (b) (c) (d)

Figure 3.2: Wrist and forearm segmentation using the depth

This segmentation was found experimentally while being constrained to lie between the forearm and wrist. Points in $\mathcal{P}$ further from the camera than W_{depth} form the forearm set defined as;

$$\mathcal{P}^W = \{\mathbf{p} \mid \mathbf{p} \in \mathcal{P}, w_z(\mathbf{p}) >= W_{depth}\} \tag{3.5}$$

The result of segmentation can be seen in figure 3.2 for several examples. It should be noted that this segmentation is likely to fail should a hand be slanted backwards. This shortcoming can be ignored giving the application requires users to present their hands in front of their body, ensuring a forward incline. An alternative method for partitioning the hand is presented in Chapter 4 which uses discriminative learning.

3.2.1 Hand Centre localisation

The centre of the hand must be estimated for tracking the hands motion, which also serves as the seed location for geodesic computation. A naive approach to localise the hand would be using the centroid of points $\frac{\sum \mathcal{P}^H}{|\mathcal{P}^H|}$ however this would result in the seed point shifting when the fingers are folded forming a fist. Therefore the chamfer distance of $\mathcal{P}^H$ is used, which considers the boundary information, inspired by Oka [75] to identify the hands centre. The chamfer distance transform computes the distance for each point in the image to its closest boundary point. The location which maximises this transform is the innermost location of the hand. The centre of the hand is defined as $\mathbf{p}^{\widehat{H}}$. This process is conceptually similar to iteratively eroding the hand, removing the finger and contour noise first until a centre is found. This process can be seen in figure 3.3 for several hand poses.

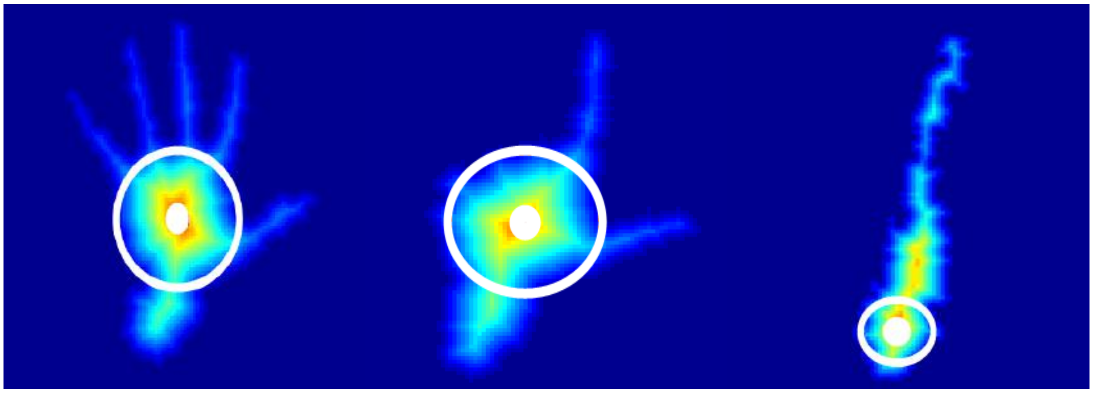

(a) (b) (c)

Figure 3.3: The chamfer distance of the segmented hand for three typical poses.

For the majority of frontal poses, this approach is robust as the palm is the widest part of the hand. However, this is not the case for hands which are profile to the camera, demonstrated in figure 3.3(c) where the most inner point moves towards the wrist. However, this is unlikely to cause failure when searching for geodesic maxima as the fingertips are still relatively far from this point.

The centre of the palm is then used to find candidate fingertips by performing a search for geodesic maximum discussed in more detail in the following section.

3.3 Candidate Finger Detection

The fingertips can be considered extremities of the hand when extended, which is consistent with the use case for Multi-touchless tracking. Through mapping the surface of the hand and searching for surface extrema, those fingers which are not folded can be found. The approach does not consider those fingers which are folded as folded fingers are not considered important for Multi-touchless interaction. This is analogues to multi-touch screens disregarding fingers, not in contact with the screen. The search for extrema is conducted by computing the minimal geodesic distance from the centre of the hand to all other points on the surface of the hand. This is performed using Dijkstra's shortest path algorithm. Those of the greatest local value correspond to geodesic maxima. Theoretically, a search for only five extrema would account for all open fingertips. In practice, the wrist can also form additional extremities that are of similar geodesic magnitude. These additional extrema are filtered in Section 3.4. For this reason, we greedily compute multiple extremities, ensuring each finger is accounted for. The extremity associated with a finger which is actually folded forms an additional false-positive to be filtered at a later stage. Those fingers which are extended yet touching are only considered a single extrema which is addressed in Chapters 4 and 5.

In order to find multiple geodesic extrema we first build a weighted undirected graph $\mathcal{G}$ of the hand. The points in $\mathcal{P}^H$ labelled in Section 3.2 are repersented as a vertex $v \in \mathcal{G}$. Utilising the image domain, each vertex v is connected to all neighbours u in the 8-neighbourhood of v using an edge s_{uv}. The cost of s_{uv} is computed using the Euclidean distance defined in world coordinates as $s_{uv} = ||w(u) - w(v)||$.

The evaluation of Dijkstra shortest path is performed by growing outward from the seed position $\mathbf{p}^{\widehat{H}}$, computing the minimal distance to each of the nodes visited. As such, each vertex in $\mathcal{G}$ stores its minimal surface distance to $\mathbf{p}^{\widehat{H}}$. During growth, each newly visited node v stores a reference to the neighbour u that came before it, allowing a path to $\widehat{H}$ to be defined. Initially, the distance to each node is of a high positive value. Should a path leading to a node offer a shorter route then it is stored as the improved route. Once the geodesic distance has been computed to all points, growth is terminated and the node with the largest distance is considered the geodesic maximum

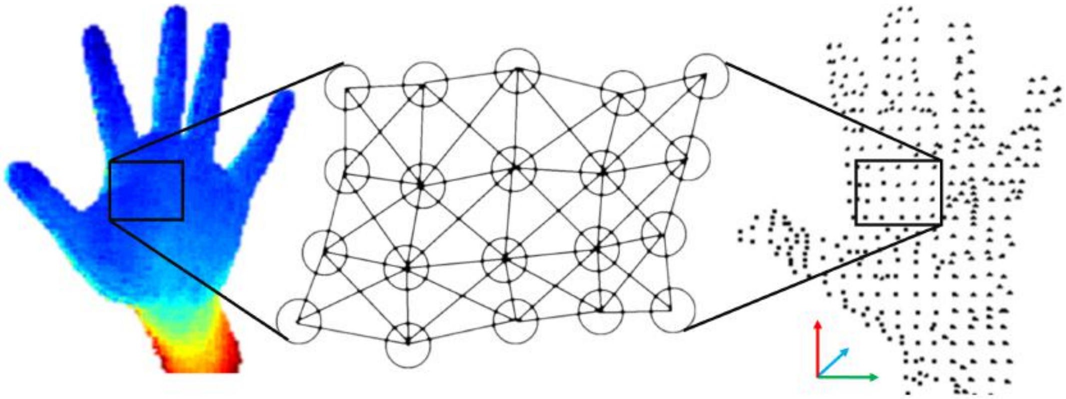


Figure 3.4: The graph construction using the image coordinates to form an 8-neighbourhood undirected connection between pixels. The edge costs are derived from the Euclidean distance between the pixels 3D location, found using intrinsic calibration of the sensor.

$\mathbf{e}$ or the longest, shortest path. An iterative search for geodesic maximum is then performed through repeated evaluations of the shortest paths to each node. To ensure that the same extrema are not revisited, two approaches can be performed. The first proposed by Plagemann [81] introduces a zero cost edge between the seed position and the previous extrema as $s_{\widehat{H}\mathbf{e}} = 0$. This presents a short-cut between the centre and fingertip candidate which ensures a distance of zero inturn leads to new longest-shortest path.

An alternative approach for searching multiple extrema was introduced by Baak [6]. By using a non-initialised distance map, repeatedly performing Dijkstra's shortest path with zero cost edges incurs a large computational inefficiency. This is because in each successive search, those nodes which are geodesically closer to the seed than the previous extrema, will maintain the same shortest path, hence reinitialisation of the distance map, leads to every node being revisited. However, this needn't be the case as the computation of nodes surrounding each extrema is unlikely to lead to a new maximum.

The search for the first extrema is performed in the same way as Plagemann's approach [81]. However, the distances computed between each successive search are preserved. The search for extrema is then repeated, using the previous extrema as the seed position. The shortest path to each extrema is computed as before. On the approach to a previous extrema, the growth is prevented, due to it incurring a larger cost than the route leading away from the previous search. This forms an equilibrium such that growth avoids prior extrema, reducing the number of nodes traversed. This allows new extrema following the first search to be found with reduced computational expense.

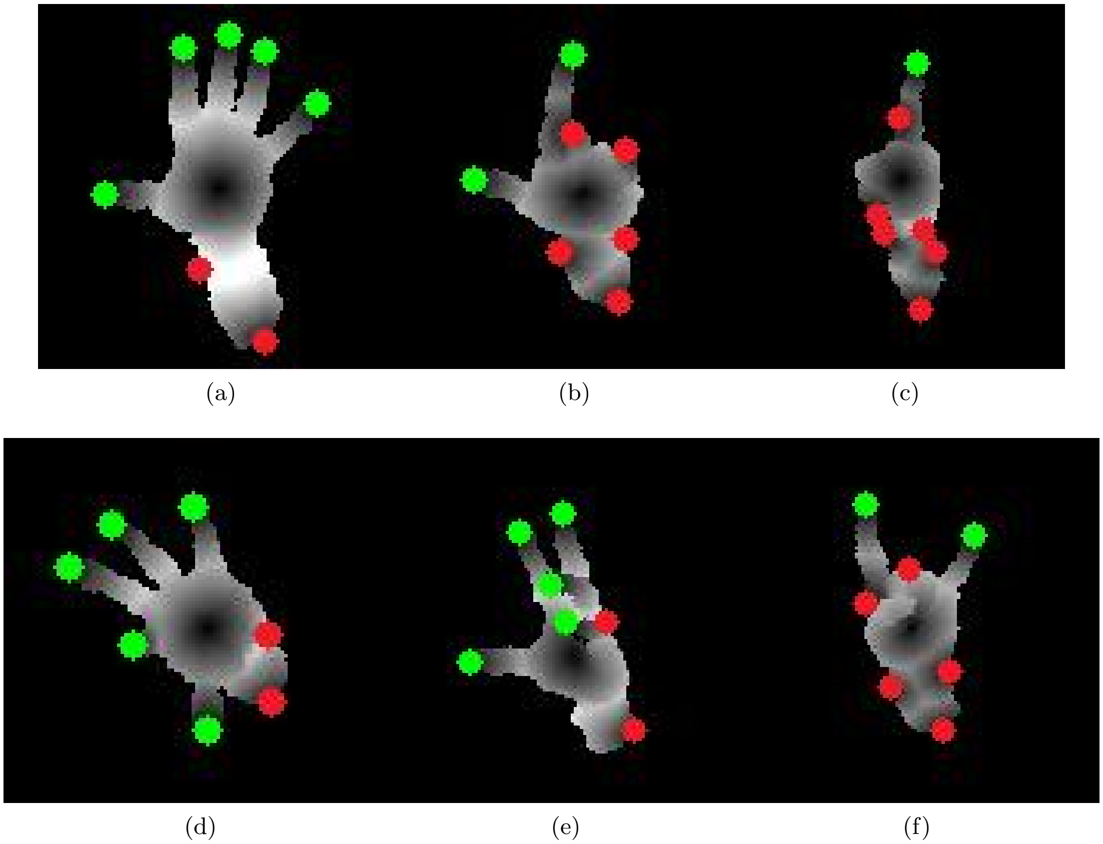


Figure 3.5: Iterative search for geodesic extrema across a range of poses, typical for interaction. The green landmarks identify correct extrema. Red landmarks represent those which are filtered in Section 3.4. The example in e demonstrates the approaches ability to localises extrema that would not be found using contour based approaches

Performing graph based distance analysis utilises the surface structure of the hand. Finding the extrema in this way is robust to contour noise which is frequent in the depth image. Such noise would cause contour-based features to fail leading to missed detections. Extrema are found iteratively using Dijkstra's algorithm with a non-initialised distance map, reducing the computational intensity. The seven extremities found for various hand shapes are shown in figure 3.5. The first seven extremities are defined as $\mathcal{E} = \{\mathbf{e}^1, ..., \mathbf{e}^7\}, \mathbf{e}^i \in \mathcal{P}^H$. Each extrema is associated with its shortest path, shown in figure 3.7. Formally, the path for the i^{th} extrema ($\mathbf{e}^i$) is defined as an ordered set of vertices $\mathcal{V}^i = \{\mathbf{v}_1^i, ..., \mathbf{v}_{|\mathcal{V}^i|}^i\}$ where, $\mathbf{v}_1^i = \mathbf{e}^i$ and $\mathbf{v}_{|\mathcal{V}^i|}^i = \mathbf{p}^{\widehat{H}}$. The false positives must then be identified and removed.

3.4 Non-Fingertip Rejection

Once the set of fingertip candidates $\mathcal{E}$ have been found, there are upwards of two false positive extrema which may be due to folded fingers and wrist extrema. Those which do not belong to fingertips must be filtered to a subset of valid fingertips. There are a number of approaches that can be performed, template matching has been used in the past [81] however, the fingertip can have a wide range of appearances due to finger orientation, which is challenging to normalise, given the limited resolution. There are also increased challenges due to the contour noise, inherent to depth sensors that would lead to curvature based approaches being unstable. Optimisation of a kinematic model offers robust localisation [6,59] but with increased computational complexity. However, the route of the shortest path across the hand provides insight. The proposed approach considers the path taken to each extrema location relative to the wrist. It is highly unlikely that the path should approach the wrists location, due to kinematic limitations. For frontal interaction it is reasonable to assume the fingers are projecting towards the camera. Therefore a path that traverses towards the camera with decreasing depth is favourable. A penalty metric is formulated that considers both of these criteria. This is highly efficient to compute allowing four hands to be processed simultaneously.

The penalty criterion seeks to remove falsely identified tips that reside around the wrist. The first stage is to localise the wrist using the segmentation computed in Section 3.2. The wrists centroid $\mathbf{p}^{\widehat{W}}$ is determined using the centroid of the set of points $\mathcal{P}^W$,

$$\mathbf{p}^{\widehat{W}} = \left(\frac{1}{|\mathcal{P}^W|} \sum_{\mathbf{p} \in \mathcal{P}^W} c(\mathbf{p}) \right), \tag{3.6}$$

Using the centroid is robust, as there is little variance in the shape of the forearm due to limited articulation, unlike the hand. To calculate the penalty score, a coarse model of the hands shape is used to represent the hand pixels $\mathcal{P}$. The model must be fast to compute while being sufficiently accurate in order to be robust to changing pose. This must include global rotation and hand poses ranging from a closed fist to splayed fingers. Using the covariance of the points $cov(\mathcal{P})$ an ellipsoid shape is used to model the overall shape of the hand and forearm. This considers global orientation and is tolerant of under segmentation from the background. This robustness to over segmentation can be seen in figure 3.7.

The covariance $cov(\mathcal{P})$ is found using,

$$cov(\mathcal{P}) = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{p}\in\mathcal{P}}(c(\mathbf{p}) - c(\mathbf{p}^{\widehat{H}}))(c(\mathbf{p}) - c(\mathbf{p}^{\widehat{H}}))^T \tag{3.7}$$

where $\mathbf{p}^{\widehat{H}}$ is the palms center.

The covariance forms a masking ellipse centred around the wrist. Pixels that have a Mahalanobis distance within three standard deviations (variance of 9) of the wrist are marked as 1, while pixels outside of this boundary are marked as 0, which is shown in Figure 3.6. Three standard deviations is chosen so that the ellipse encompasses most of the hands area. The following equation details the formulation of this mask,

$$M(\mathbf{p}) = \begin{cases} 1 & \text{if } (c(\mathbf{p}) - \mathbf{p}^{\widehat{W}})^T cov(\mathcal{P})^{-1}(c(\mathbf{p}) - \mathbf{p}^{\widehat{W}}) < 9, \\ 0 & Otherwise. \end{cases} \tag{3.8}$$

Using the path from the hand's centre to each extrema, and the elliptical mask, the penalty $S(\mathcal{V})$ can be found. The paths score is incremented for each vertex with increasing depth through the masking ellipse and then normalised using the complete path's length. It is important to note that when moving along this path, a vertex is only included in the penalty if the current vertex's $\mathbf{v}_i$ depth is less than the next $\mathbf{v}_{i+1}$ ie, $w_z(\mathbf{v}_i) < w_z(\mathbf{v}_{i+1})$. This term is consistent with the understanding that the wrist has a greater depth than the centre of the hand, hence we only consider vertices that traverse with increasing depth. This is found for the path associated with each candidate in $\mathcal{E}$

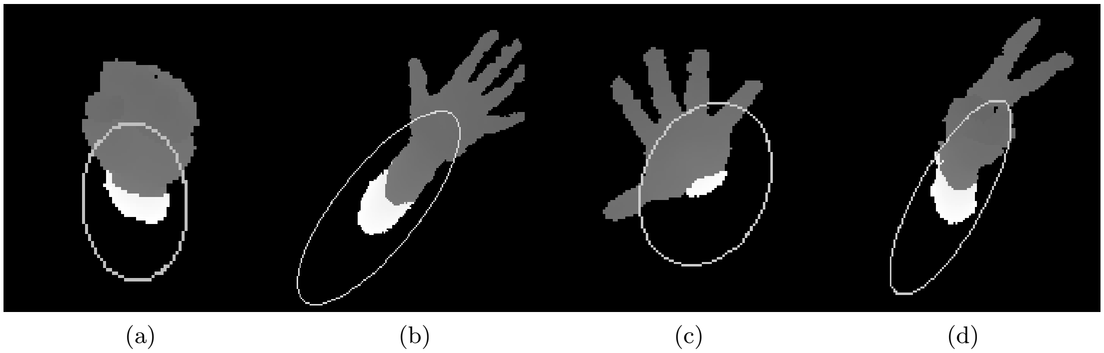


Figure 3.6: Ellipse formed around the wrist using the covariance of the hands and forearm points

with the following;

$$S(\mathcal{V}) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|} M(\mathbf{v}_i)\mathbb{I}[w_z(\mathbf{v}_i) < w_z(\mathbf{v}_{i+1})] \tag{3.9}$$

Figure 3.7 shows the vertices in red that contribute to an increased score.

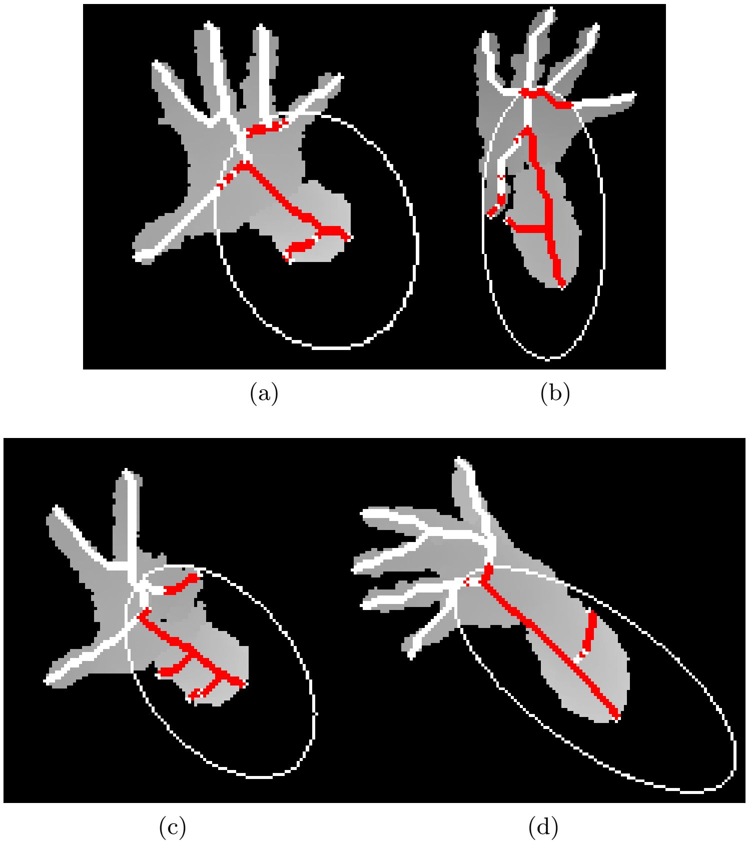
The candidates are then filtered using Equation 3.10 down to a subset of candidates $\mathcal{E}' \in \mathcal{E}$, that exclude wrist extrema, where the value of θ is determined using parameter tuning detailed in Section 3.5.

$$\mathcal{E}' = \{\mathbf{e}' : \mathbf{e}' \in \mathcal{E}, S(\mathcal{V}) < \theta\} \tag{3.10}$$

Instances of folded fingers lead to additional geodesic extrema that are incorrectly located. These form around the palm of the hand and must be reduced using additional criteria. A sphere that encapsulates the contour of the palm is used to eliminate those extrema that are found within its radius. Extrema detected inside of the sphere are excluded from interaction.

The Euclidean distance of the fingertip, to the centre of the hand, is used to reject extrema located around the fist. The remaining fingertip extrema are formalised as, $\mathcal{F} \in \mathcal{E}'$ where,

$$\mathcal{F} = \{\mathbf{f} : \mathbf{f} \in \mathcal{E}', d(w(\mathbf{f}), w(\mathbf{p}^{\widehat{H}})) > \beta\}, \tag{3.11}$$



(a) (b)

(c) (d)

Figure 3.7: The complete shortest path for each extrema. The points along each path that contribute to an increase in the score are coloured in red. (a) shows that vertices outside of the ellipse do not contribute to the penalty. The path of the thumb in (b) demonstrates the need to only penalise vertices that traverse with increasing depth. (c and d) demonstrate additional pose variation.

given the distance $d()$ is the L2 Norm and β is the spherical radius. This hard cutoff was chosen so as to give a consistent condition of when a fingertip is detected, improving user interaction. The value of β was chosen using parametric tuning across multiple users and multiple hand shapes, which is detailed in Section 3.5.

The remaining extrema are the final detections of fingertips. These can be used as detection on a per frame basis, or used to update positions in a tracking framework. This allows the incorporation of temporal information leading to smoother performance, with reduced jitter between frames.

3.4.1 Finger Assignment and Tracking

A Kalman filter is used to track and perform assignment of the fingertips between frames. This model is updated using the point correspondence that minimises the change between consecutive frames. We found the need to check all possible permutations when matching points, as our tracking is performed in three dimensions. When assigning five or fewer points, searching all permutations ($O(n!)$) requires fewer operations than using the Hungarian algorithm ($O(n^3)$). As the hand is limited to five fingers it consists of only 120 permutations in the worst case.

The world position of each detected fingertip $w(\mathbf{f})$ is used to update a bank of three-dimensional Kalman filters $\mathcal{K}_t$ at time t. To update this model, points from $\mathcal{F}$ are paired with the predictions of $\mathcal{K}_{t-1}$ by selecting the assignment between points that have the lowest cost. This requires an indexing set to map from the predicted locations to the estimated fingertip position. This map is constructed using the permutations of the smaller of these two sets ${}^{\mathbb{N}^0}\mathcal{P}_{|\mathcal{A}|}$ where, $|\mathcal{A}| = min(|\mathcal{F}|, |\mathcal{K}_{t-1}|)$. This forms a set where each row represents one of the possible permutations. Iterating through each permutation we build the assignment set $\mathcal{A} = (a_1, ..., a_{|\mathcal{A}|}) \in {}^{\mathbb{N}^0}\mathcal{P}_{|\mathcal{A}|}$ where a is the index used to associate a fingertip to a Kalman filter. The final correspondence between fingertips is the permutation $\mathcal{A}'$ which has the lowest sum of squared differences from $\mathcal{F}$ to the indexed predictions $\mathbf{k}_{ai}$, as shown here;

$$\mathcal{A}' = \arg\min_{\mathcal{A}} \sum_{i=1}^{|\mathcal{A}|} (w(\mathbf{f}_i) - \mathbf{k}_{ai})^T (w(\mathbf{f}_i) - \mathbf{k}_{ai}) \tag{3.12}$$

Any Points in $\mathcal{F}$ that are not matched initialise a new point $\mathbf{k}$ that is introduced into the model. This is to account for the appearance of unfolded or occluded fingers. For predictions derived from $\mathcal{K}_{t-1}$ that were not matched, their Kalman filter is updated using the previously predicted position. This blind update is performed on the condition that the prediction's confidence does not diminish considerably, at which point it is removed from the model.

The Kalman smoothed model can then be used to output each of the fingertips as a three-dimensional coordinate in millimetres. These coordinates can also be projected

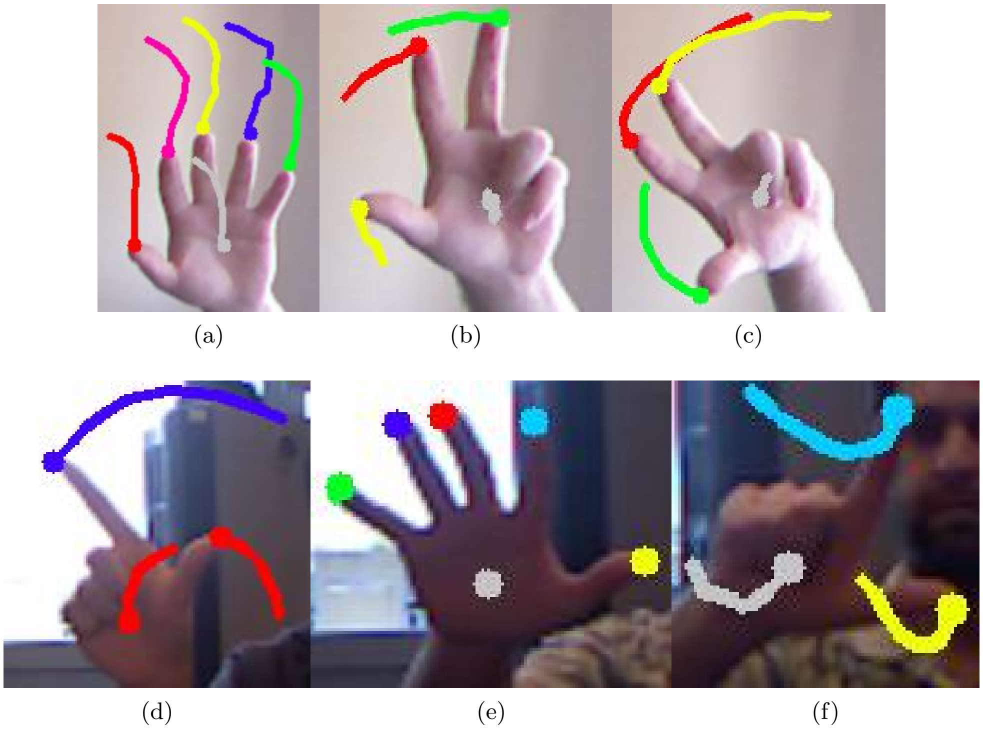


Figure 3.8: The tracking of multiple fingertips and the centre of the hand. The path followed over the previous 10 frames is drawn for each tracker. The length of each path illustrates the robust tracking during rapid motion. (d, e, f) are examples captured in the dataset, demonstrating the harsh lighting which would cause a non-depth based approach to fail.

back to the image domain for the purpose of visualisation, displaying the points as shown in Figure 3.8. It is important to track the points in three dimensions as the addition of depth allows for sub-pixel accuracy. While it is possible to track the points in the image domain, the resulting fingertips are prone to jitter due to the quantisation of the image.

3.5 Experiments

In order to evaluate performance, it was necessary to capture our own dataset, as current hand datasets are more related to pose classification. While it would have been possible to label an existing dataset manually, for example, Rens [132] gesture set, they do not contain temporal information regarding transitions between hand shapes. For our dataset, we captured ten sequences of depth. The data is captured using five adult seated users performing multiple actions, with depth ranging from 0.68m to 1.02m. Table 3.1 demonstrates the range of depths and hand sizes between the 10 adults, when normalised to a distance of 1m. The hands sizes are similar for the majority of the users tested with average size at 1m being 46.5 pixels with 3.2 σ. The frame count is also provided for each user, with a total of 7994 test frames across the user sequences.

Each user was asked to perform their first sequence with a constrained rate of movement, to assess the generalisation across users hand shapes. They were then requested to perform actions at a faster rate. This was to replicate more realistic movement that is suitable for interaction, which included transitions between gestures and natural resting poses. In particular user 6 performed several very rapid dragging gestures, and in one case moved over 140 pixel in just 8 frames. Another set of sequences consisting of varying ASL shapes were captured for parametric tuning. The ground truth was manually annotated for each of the fingertips in three dimensions. Labelling was assisted by selecting the nearest point on the hand to the point clicked in the image.

User	Depth (m)	Hand Size(Pixels)	Size 1m (Pixels)	Frame Count
1	0.95	55	52.25	1092
2	1.02	44	44.88	540
3	0.78	60	46.8	1133
4	0.75	62	46.5	540
5	0.68	68	46.24	1108
6	0.91	53	48.23	502
7	0.82	53	43.46	1154
8	0.63	70	44.1	466
9	0.85	60	51	446
10	0.84	50	42	1013

Table 3.1: Statistics regarding each user sequence in the Multi-touchless dataset.

3.5.1 Wrist Exclusion

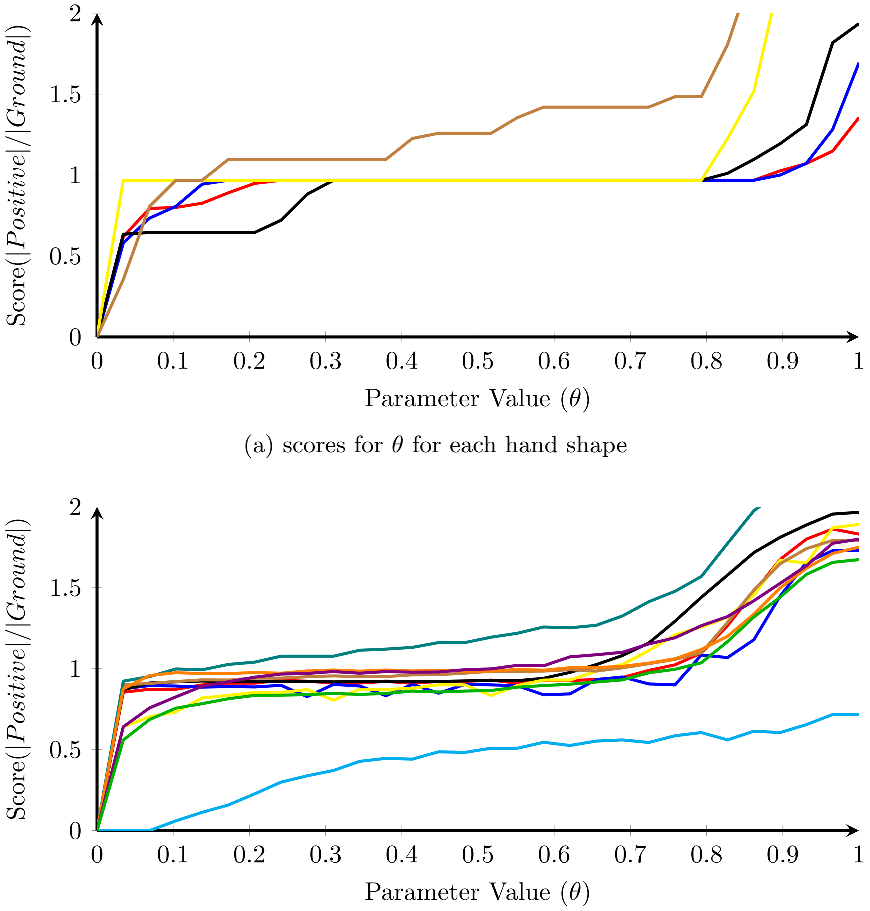
To assess the effect of the parameters θ and β, performance was evaluated over a small set of sequences before evaluation in Section 3.5.3 on a larger unseen testing set. For wrist exclusion, the parameter θ can have a possible range between 0 and 1. As the value of θ increases, filtering of wrist points relaxes, and more false-positives are detected. The ratio of fingers detected over the number of ground labelled fingers was recorded. Figure 3.9 shows the ratio of detections over ground truth track points vs θ for various hand shapes and users. It can be seen that there is a broad plateau where the value of θ does not affect performance, showing that θ generalises well across both hand shapes and users. We selected a value of 0.3 from within this region for use in our experiments.

3.5.2 Fist Exclusion

Using the same principle for fist circumference, β was tested across multiple hand shapes and users. Figure 3.10 shows a wide plateau over which parameter selection does not affect performance. We selected a value of 7mm for our experiments.

3.5.3 Evaluation

There are two main considerations when assessing the performance of the approach: the number of fingers detected with relation to the correct amount, and the precision of



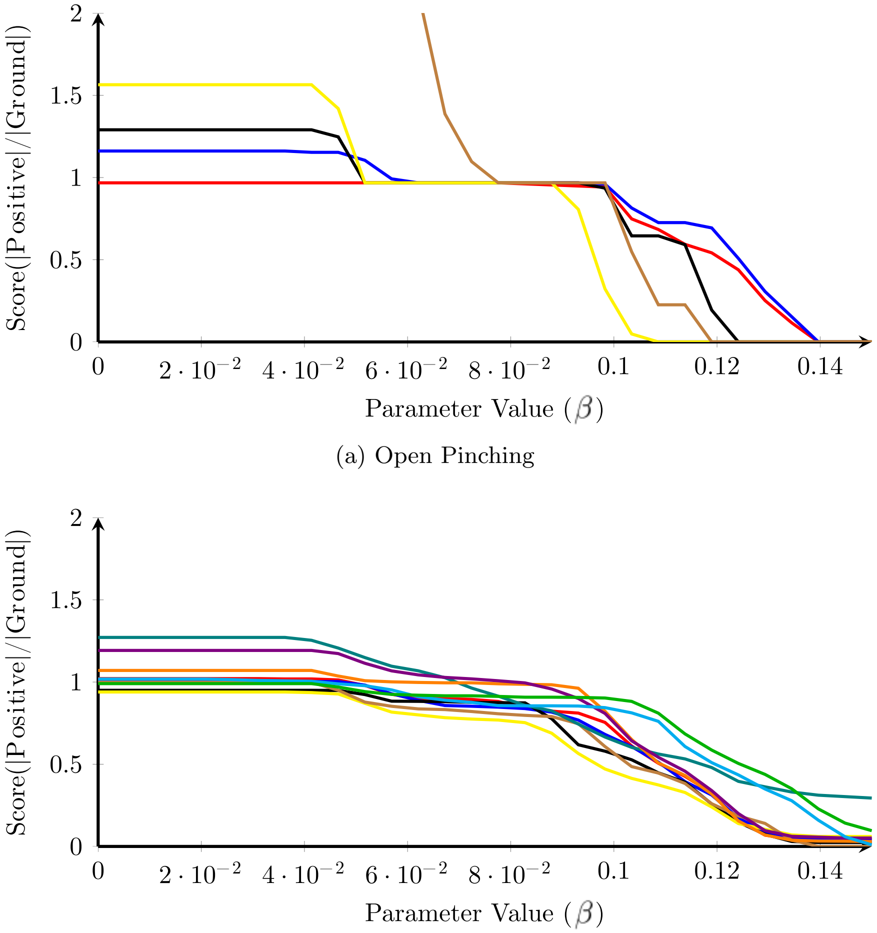
(a) scores for θ for each hand shape

(b) scores for θ for each user sequence

Figure 3.9: The results show parameter tuning of the variable θ. (a) Hand shapes in the counting sequences are plotted. (b) Results across each user, where the lower line is for sequence $5a$ where the hand segmentation is poor.

the estimated position. Combining these metrics to quantify the overall performance of the approach is to be avoided, as an appropriate weighting depends on the application. For this reason, we represent the performance of detection and accuracy independently. Correct fingertips are identified as being within (1cm) of the ground truth fingertip. The results of this can be seen for both sequences for each user in Table 3.2.

We found that there is varying difficulty across each of the sequences, with the most challenging being user 6. In this particular sequence, the user used very rapid movements for interaction. This motion caused extensive blurring in the footage, and the structure of the hand was completely lost for many frames. Several examples of these



(a) Open Pinching

(b) Self Occlusion

Figure 3.10: The results recorded while parameter tuning the variable β. (a) Hand shapes in the counting sequences. (b) The results across user sequences, and show stability for a range of values.

challenging examples can be seen in Figure 3.11.

The average error across all sequences for each hand is 24.8mm. This poor average is attributed to outliers residing around the wrist. The large percentage of landmarks within 1cm of the ground truth confirms the bias in presenting the mean accuracy. This is confirmed when observing the histogram of all errors per finger, which is shown in Figure 3.12.

A comparison with existing work is challenging as there is limited availability of annotated datasets. Our detection performs, with 80% of detections being within 5.7mm. This demonstrates that our approach can localise the fingertips accurately when com-

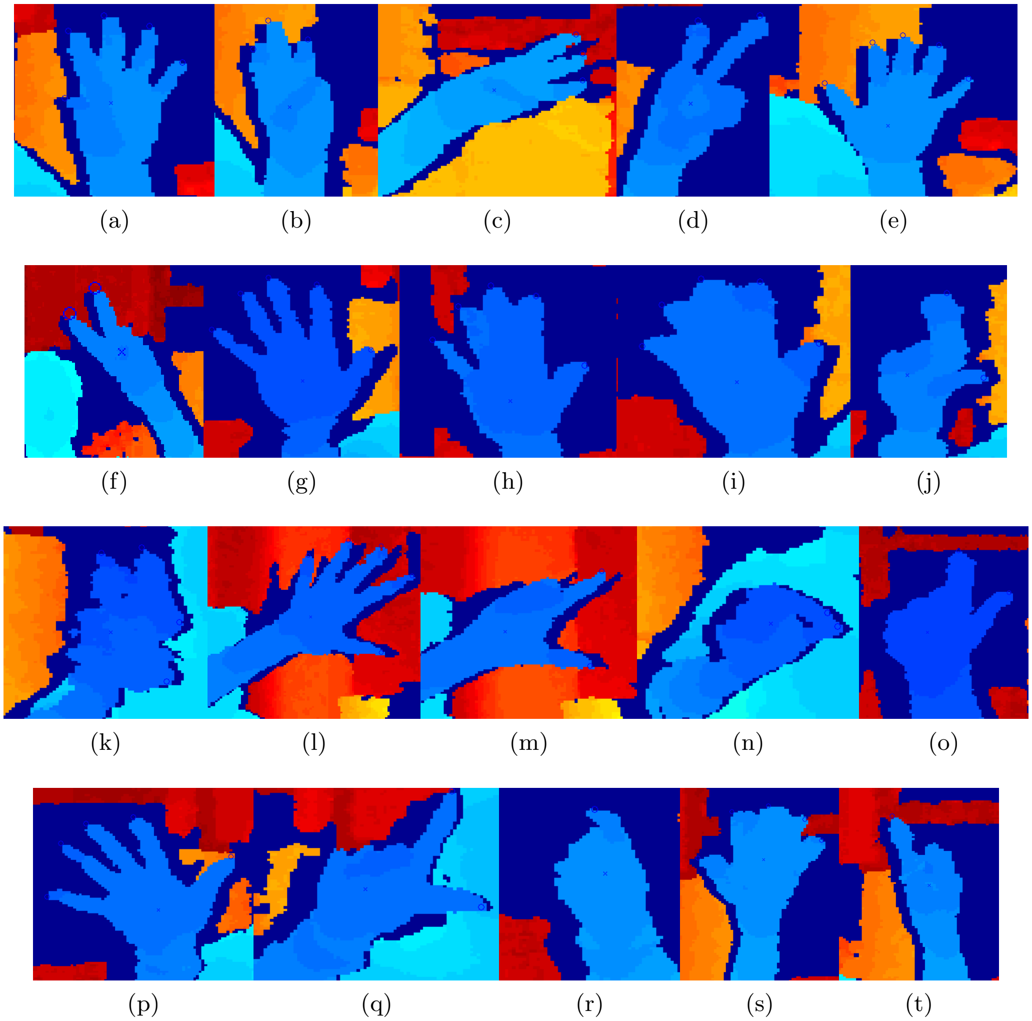
User	Error(mm)	TP(%)	TP(σ)
1	22.0	75.62	20.64
2	11.7	83.15	16.01
3	16.0	88.37	19.79
4	26.5	77.45	21.49
5	30.5	69.51	24.25
6	46.8	75.63	22.59
7	35.7	70.74	23.65
8	8.3	87.62	18.92
9	33.4	74.40	17.54
10	22.6	79.78	21.02
Average	25.3	78.23	19.11

Table 3.2: The errors for each sequence in mean mm error and the percentage of fingers that reside within 1cm of the ground truth landmarks.

pared to the size of the fingertip and does not rely heavily on temporal smoothing. Our approach concentrates on fingertip detection allowing 30fps for multiple hands without GPU optimisation reducing the need for additional hardware. Allowing the use of fingertip tracking in low-cost computing application.

3.5.4 Limitations

We found the sensor had inherent drawbacks due to being designed for the primary use case of full body pose estimation. For example the Kinect had difficulty mapping the surface of the hand when fingers are directed towards the sensor. This forms a hole in the depth information which our approach does not account for. For this reason we avoided gestures where the users points at an object on screen. Additionally in Section 3.4 we define the use of a threshold which is parametrically tuned to be invariant to adult users, however in the event of a child using the system, it might fail due to the smaller hand size. Therefore a more appropriate value could be derived at run time from the distance transform in Section 3.2.1. The max value at the centre of the hand once normalised by the depth would provide the palm width.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

Figure 3.11: Example poses from the Multi-touchless evaluation dataset, showing a range of interaction poses. The set includes rapid motion which causes motion blur.

3.6 Application of Multi-touchless

An example application of Multi-touchless interaction was in the *Making Sense*[1] project which provided an intuitive understanding of large complex datasets. Offering an intuitive interface to complex machine learning algorithms for data analysis. Figure 3.13 shows the system in use. Using data mining and machine learning tools with such an interface allowed patterns to be discovered that relate images, video and text, allowing an analyst to quickly:

[1] http://www.making-sense.org

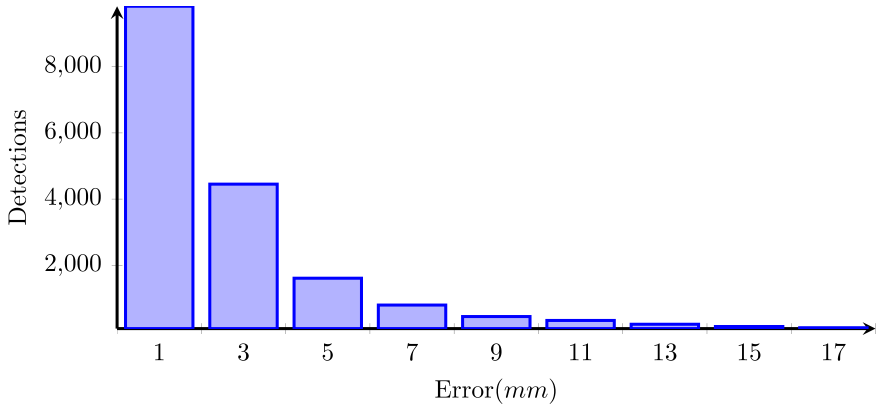


Figure 3.12: A histogram of the detections across all user sequences, showing the majority of detections are within 5mm of their ground truth.

- Visualise data
- Summarise the data or subsets of the data
- discriminatively mine rules that separate one set of data from another
- Find commonality between data
- Project data into a visualization space that shows the semantic similarity of the different items
- Group and categorise disparate data types

Using data mining approaches, analysts were able to perform interaction with a timeline of observed events. These events represented a captured users interaction with large quantities of online multimedia data. Using Multi-touchless the analyst was able to search the timeline for recurring events and trace a user's interaction. An overview of the Making Sence system can be seen in figure 3.14.

The Making Sense visualisation system consists of two displays: a curved projection screen which forms the main display and a secondary table display (see Figure 3.14 for an overview). Two HD projectors are pre-distorted to correct for the curvature of the screen and stitched together to form a single 2500x800-pixel display used for displaying time-line data. As the user stands at a distance from the screen, conventional means

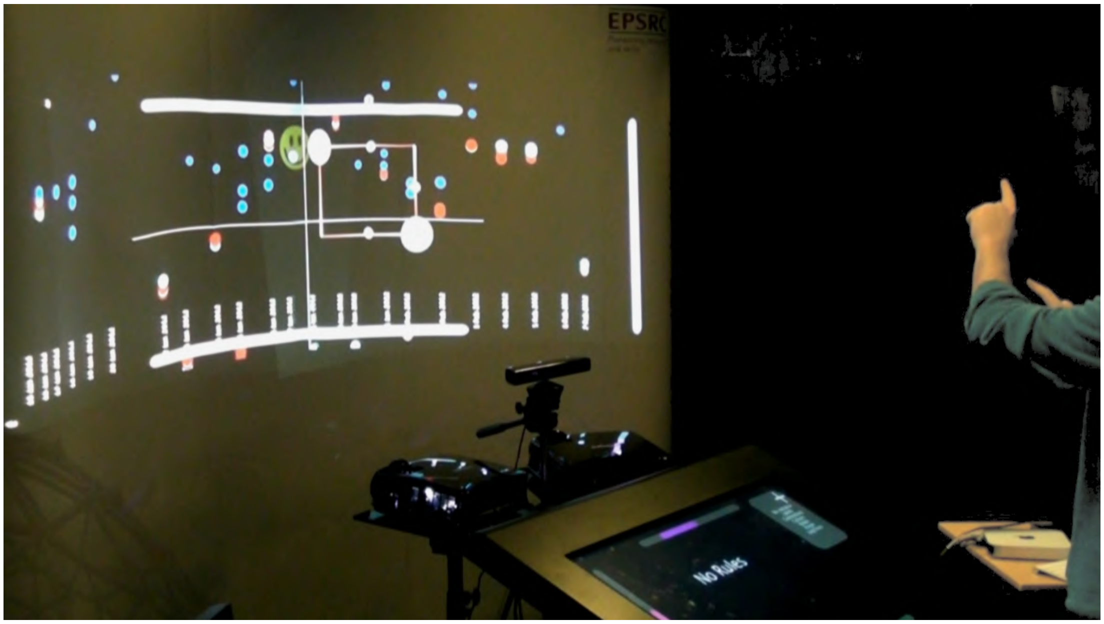


Figure 3.13: Interaction with a parabolic display for data mining. This is only possible through the use of a Multi-touchless interface. Conventional computer interaction such as using a mouse and keyboard would not be applicable. Credit: [48]

of interaction such as a mouse and keyboard are not applicable/required. The system, therefore, employs a gesture vocabulary discussed in Section 3.6.1.

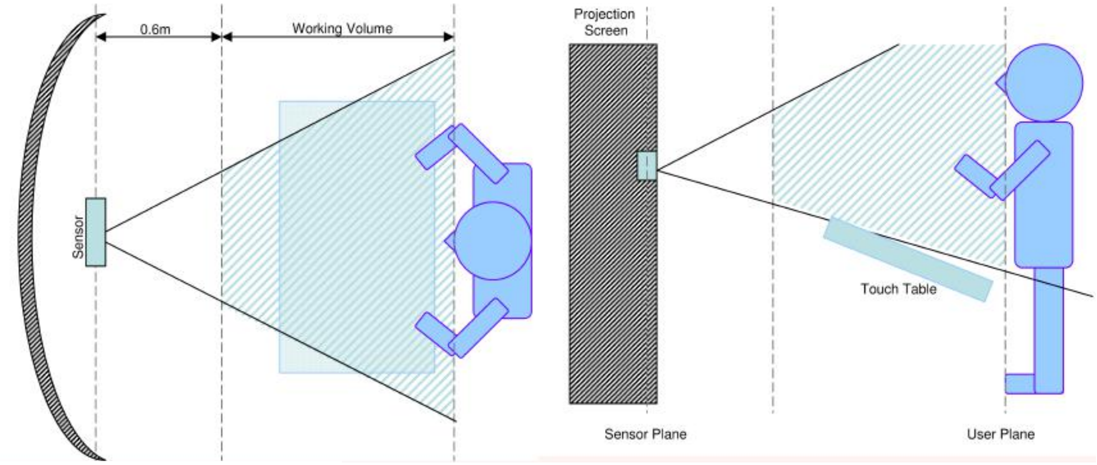
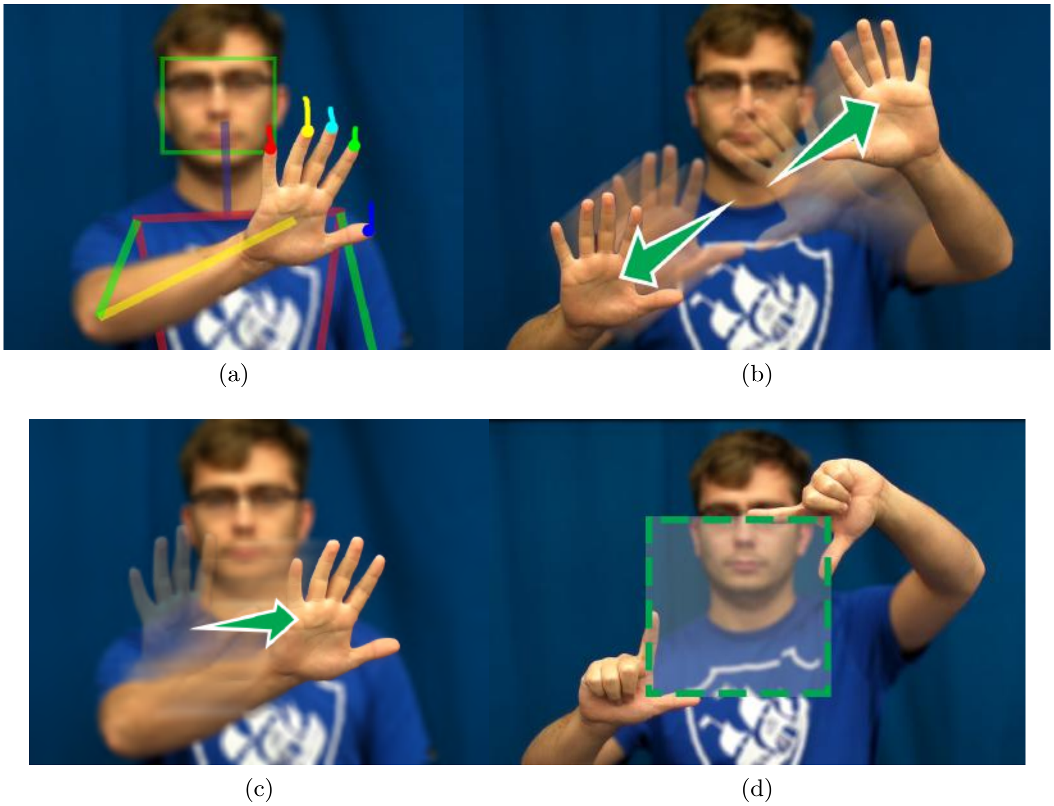


Figure 3.14: Overview of the Making Sense system. Credit: [48]

A second table top display directly in front of the user employs a traditional multi-touch overlay on an HD display forming a traditional multi-touch table. However, as the space above the table is within the operational volume of multi-touchless recognition, gestures can be used to interact with both the projection screen as well as the table by looking at the direction of the users arms.

3.6.1 Recognising Gestures

Gestures for interaction are typically common across multi-touch devices and provide a familiarity that allows the user to perform complex tasks with ease. However, interacting in 3D is a fundamentally different experience and 2D gestures do not necessarily translate well to 3D. Consideration must be given when developing gestures, as they are integral to the user's experience. Gestures should be simple to perform and be semantically similar to the task they represent; a good example would be pinching to zoom. However, for a large-scale display, a single-handed two-finger pinch/zoom does not scale well to the size of the visualisation and a two-handed, single-finger zoom is more appropriate.


(a)

(b)

(c)

(d)

Figure 3.15: Examples of zooming, swiping and selection gestures implemented in the middle-ware.

To provide flexible multi-touchless gesture recognition, middleware sits between the finger tracking level and application level. The gesture vocabulary is chosen to generalise across applications. To minimise the complexity a new user experiences with a change in the interaction modality, we chose the most common 2D gestures for adaptation.

From hand and finger tracking, we know the position and presence of a user's head, hand and fingers as shown in Figure 3.15a and additional skeletal information is also available from the Kinect. We use a representation of a hand's velocity to detect both horizontal and vertical swipes (Fig. 3.15c) both single or double handed or a zoom (Fig. 3.15b). As we know the number of fingers present on each hand, these gestures are activated for specific configurations of fingers. For example, at the application level we detect swipes regardless of the number of fingers present, but used one finger per hand to activate the zoom gesture. However, other combinations can be used for additional functionality. For example, a commonly used gesture is an extended thumb and index finger to control a virtual cursor on the screen, selection can then be achieved by retracting the thumb to select the object under the index finger. Grasping actions are also easily detected as the rapid transition from five to zero fingers.

Figure 3.15d shows a selection window gesture which directly translates to a variable size selection window in the application layer. Its size and position are controlled by the position of the hands. This is a particularly intuitive gesture that we use in the demonstration system for selecting different sets of data. As the hands are tracked in 3D we can also use rapid motion towards the camera to identify push gestures. A push gesture with an open hand can be used to open a menu system or push combined with a single index finger can be used to select an item. Rotation of the hand can also be used for control of polar menu systems. However, further combinations can be used to extend the vocabulary.

More complex gestures that involve transitions, for example, a horizontal swipe followed by a downward swipe could be better represented using a probabilistic approach such as Hidden Markov Models.

3.7 Conclusions

This chapter presented a Multi-touchless approach for interaction. Providing a contactless interface, the Multi-touchless system allows new applications of computing, where conventional touch screens are not applicable. Tracking the fingertips in three-dimensions can be integrated with existing multitouch applications while offering an additional dimension for gesture design.

Posed as a graph-based approach, fingertip detection can be performed efficiently for a number of hands in real time. By representing the hands surface as a graph, an iterative search for geodesic extrema was performed using an efficient application of Dijkstra's shortest path algorithm. The search included additional extrema ensuring robust detection of all of the fingertips. Performing the search with depth improves performance against contour based approaches that face challenges regarding contour noise and occlusion.

Additional extrema not attributed to fingertips were removed. Those extrema close to the wrist were filtered using structural information. The method consisted of a penalty criterion which observes the traversal path leading to each extrema. Those which traversed close to the wrist were excluded from interaction. The wrist itself was modelled using an elliptical model, ensuring invariance to poses with global rotation and varying degrees of articulation. Candidate fingertips are then tracked using a Kalman filter to provide temporal smoothing and correspondence between frames.

Evaluation of the approach was performed against a large dataset consisting of several users. The sequences captured were challenging with ranging poses, and fast paced motion. Tracked fingers were compared against manually annotated landmarks, demonstrating the accuracy of localisation in terms of millimetres. The approach was compared with that proposed by Oikonomidis [72]. While their approach is significantly different from ours, they quote their performance in-terms of millimetre accuracy. This shows comparable results in-terms of fingertip localisation with limited computing resources, allowing integration with mainstream computing.

Using a structural approach such as this has two main shortcomings. Firstly the method is not able to identify the fingertips. This means there is no distinction between the thumb and little finger's tip. Secondly the approach is unable to directly localise less salient regions of the hand, such as the root joint of each finger. The following chapter investigates an alternative approach that utilises machine learning.

Chapter 4

Hand Pose Estimation using Randomised Decision Forests

This chapter investigates the use of discriminative modelling for hand pose estimation. In particular, the training of a RDF to perform classification of the hands regions.

Discriminative methods such as Randomised Decision Forests (RDFs) have become popular in body pose estimation and have been proven to evaluate with real-time performance for hand and body pose [42, 98]. As a detection-based approach, using only a single depth image, they typically operate without temporal information. This removes the requirement for initialisation, which generative approaches [72] depend on. By not requiring reinitialisation discriminative approaches allow a more natural user experience in failure recovery.

Large datasets are required to train such discriminative methods which are challenging to acquire. The approach proposed by Shotton [98] consisted of a combination of motion capture and synthetic rendering, compiling a labelled training set of body poses. In training a model that represents the hands range of motion, one would require an even larger dataset. As a solution, Keskin [42] applied an RDF to hands that were trained entirely using synthetic data. An alternative means of training is presented in this thesis, which incorporates the use of coloured glove as a training aid. This allows colour to provide the region labels during training. This knowledge can then be transferred to the depth modality, which is consistent with an ungloved hand. The contributions of this chapter are 1) an examination of Randomised Decision Forests (RDFs) and the parameters used in their training, investigating their application to hand pose estimation. 2) The use of a coloured glove to provide an autonomous means of labelling large quantities of hand examples and the extension of a GMM classifier to reject non-distinguishable colour classes, enforces robust labelling. 3) The direct encoding of ASL fingerspelling labels in the forest structure, removing the need to perform second stage learning of hand shapes.

4.1 Randomised Decision Forests

A Randomised Decision Forest (RDF) is a machine learning approach that is capable of performing classification and regression, by identifying discriminative features. RDFs are an ensemble based method, consisting of a number of Binary Decision Trees. This use of multiple tree structures was introduced by Shlien [97] for handwriting recognition and continued with Amit et al [2] proposing the use of structural features. Breiman [11] later investigated the training mechanisms, aiming to improve the forest's ability to generalise. They have since been applied in many areas of computer vision but notably to Human Computer Interaction (HCI) by Shotton et al [98] who used an RDF to perform classification of the regions of the body. This demonstrated the forest's ability to model a large variety of data, learning from more than 300,000 pose examples. Operating on a pixel-wise basis, the body's parts are identified using depth-based features. The features sample both local and global context, and those which provide the most information are identified.

This chapter will focus on the use of a RDF due to the real time requirements and its ability to model large quantities of training data. While segmentation has been demonstrated using CNNs by Neverova [71], frame rate is bellow real time for CPU implementations. Conversely, forests are highly efficient at evaluating the depth, allowing its use as part of a framework in real time. This is attributed to the limited number of features that must be evaluated and is discussed with greater detail in Section 4.1.1.

A forest consists of multiple trees trained independently of each other. This aims to

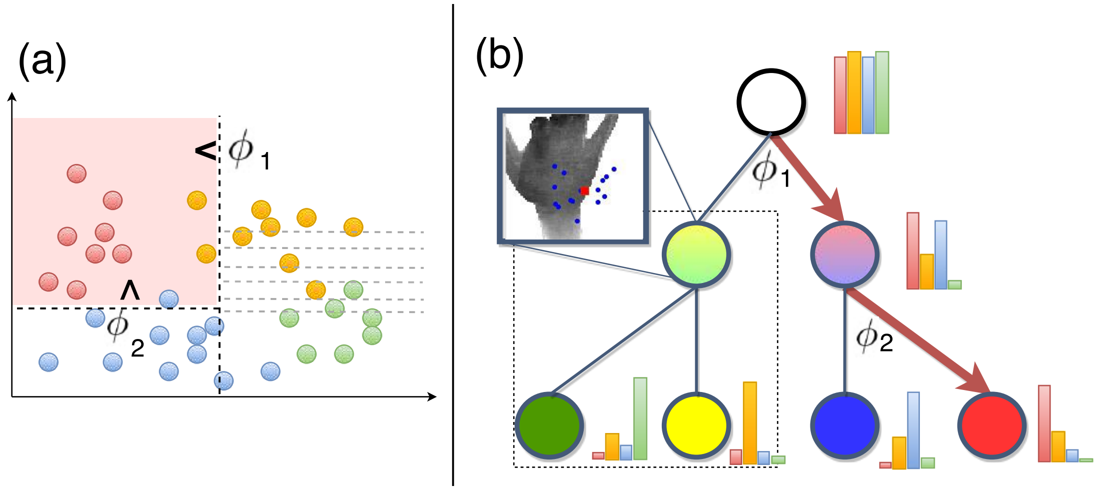


Figure 4.1: The training examples are shown as coloured points in a two dimensional space. This space is separated by the tree forming axis aligned splits (a). The tree structure can be seen where each level of the tree improves the separation of the examples (b). Features sample the depth image to find a split that partitions the remaining samples shown in (a).

improve accuracy, where each tree in the ensemble is trained over different partitions of the training set, preventing the forest from over-fitting to the training data. The trees themselves are composed of a branching structure, consisting of many nodes. This structure begins with a root node residing at the top of the tree, where test sample pixels are fed. The root node is one of many decision nodes that perform binary classification, allowing information to be partitioned as it passes down the tree. Samples passed to the root node are classified into a left and right subset. This decision of whether a sample should propagate left or right is based on features of the observation, and prior responses during training. The left and right subsets are then propagated to two child nodes which again, make a binary decision. Each child node splits this data into two and this process of partitioning continues until either the data is exhausted or a maximum depth is reached.

The structure of a tree which has 3 levels is shown in Figure 4.1. This results in increasingly smaller distributions of data with each binary decision made. Presuming discriminative features were found during training, samples will follow consistent routes through the tree at test time, allowing inferences to be made. Once samples reach the leaf nodes of the tree, the label at that node is assigned to the test pixel. For a classification tree, each leaf node contains a distribution of the class labels seen during training, which is then returned as the trees prediction. The hypothesis of each tree is then combined with other trees of the forest, forming a single, more robust prediction.

In the process of branching, each node learns a feature which separates the training data using the context of the surrounding depth. As the features are used to determine the splitting at branches of the tree, they are described as splitting criteria.

4.1.1 Splitting Criteria

In order to partition the training data, discriminative features must be determined. With depth images there is no texture information, therefore cues for a part of the hand are best represented using the contour and surface gradients.

The tree structure is capable of operating over relatively weak features as they are combined as the tree is traversed. At the higher levels of the tree rough, inaccurate partitions are made that have limited discriminative strength. However, progressively complex structures can be represented as the tree increases in depth. This is because each node builds upon the prior understanding of its parents nodes, allowing the tree to disambiguate subtle differences. With each additional level of the trees depth, more separable patterns emerge allowing very confident predictions to be made at the leaf nodes.

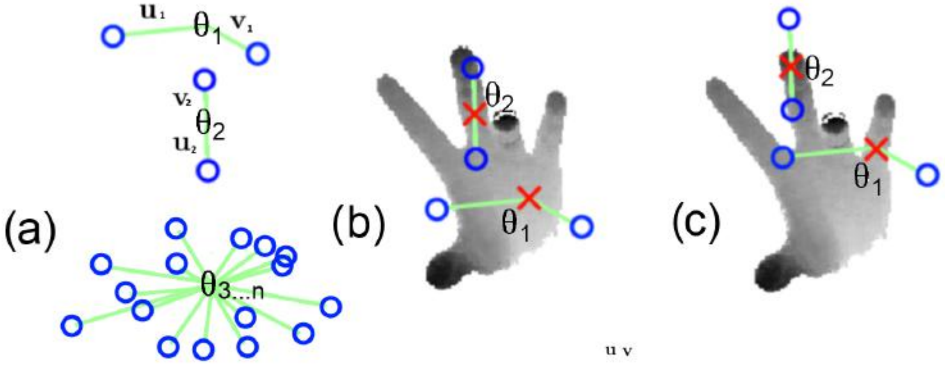


Figure 4.2: Features used to sample depth. The random offsets for values of θ are shown in (a). In image (b) the features θ_1 and θ_2 give a different response from those sampled in (c). This differing response provides discriminative features that are both translation and depth invariant.

For each pixel, two random offsets $\mathbf{u}$ and $\mathbf{v}$ are drawn from a uniform distribution. The offset vectors probe the depth image to identify local context. This offset probing can be seen in Figure 4.2, where the offsets are shown in green. This provides information regarding the depth of two locations surrounding each pixel. By sampling many combinations of random offsets and assessing their ability to separate the different classes of object, distinguishable features in depth can be found. Only those offsets which provide the most discriminative features are recorded.

During testing only two offsets must be probed, for a node to make a decision. This means evaluating a pixel requires at most two times the trees depth $(2T_d)$ pixels to be accessed from the image. This greatly reduces the number features that must be evaluated when compared against alternative methods of classification. As such, the RDF is computationally efficient at test time.

The comparisons in depth are computed over a number of random positions surrounding the sample pixel. The difference in depth is then compared against a threshold and has been used in several depth based methods [42,98,114], demonstrating its discriminative ability. The feature response for a pixel $\mathbf{x}$ is calculated as,

$$F_\phi\left(\mathbf{I},\mathbf{x}\right) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I\left(\mathbf{x}\right)}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I\left(\mathbf{x}\right)}\right), \tag{4.1}$$

where $\phi = (\mathbf{u}, \mathbf{v})$ represents the pair of offsets used to compare depth, and $d_I(\mathbf{x})$ provides the depth of $\mathbf{x}$ in its respective image $\mathbf{I}$.

In using offsets to probe the depth image, the features are translation invariant. They are also scale invariant as the lengths of $\mathbf{u}$ and $\mathbf{v}$ are normalised by the central pixel, reducing the features size as the hand increases in distance.

The locality of the features can also be adjusted. The features locality can be defined by bounding the offsets to a maximum fixed radius r_{max} during training. A large radius allows the features to sample across the entirety of the hand. However, as there is a limited number of features that can be evaluated, this increased radius reduces their density. In reducing the density, less information regarding the local configuration of the joints can be observed. This could lead to less optimal decisions in the trees, in turn reducing performance. Sampling from a small radius restricts the feature's visibility of the hand but also improves the observation of local context. This is particularly important considering the strong kinematic dependencies between the base of fingers and the fingertips (implicit hierarchical constraints). This means r_{max} is dependent on the size of the hand and the observed configurations. As such it is optimised during parameter selection in Section 4.2.2.

It should also be noted that the features are not rotationally invariant to in plane rotations. This means unique discriminative features must be found for rotated variants of the same pose. To learn such features, rotated examples of each pose candidate must be present in training. Synthetic rotation of the training data can be performed to encapsulate global rotation. This does mean that the forest must accommodate a vast range of poses but is tractable due to the branching structure. Alternatively, the features offsets can be rotated by an in-plane rotation [126]. However, this requires the addition of an initial regression stage to determine the hands orientation. Should the users skeleton configuration be available such as with the Kinect, the in-planer component can be used to normalise the hands rotation which simplifies the process.

The features also assume that the depth image of the hand is complete. This is not always the case as there are artefacts due to the acquisition by structured light. The fingertips are often too small for the light pattern to be recovered, resulting in a hole in the depth image and this can lead to misclassification. This is less prevalent in both the second generation Kinect and Intel sensor, which utilise Time-of-Flight (ToF). The hand should also be separable from the background. Interacting hands and the grasping of objects can result in depth appearance not seen during training, leading to partial misclassification.

4.1.2 Training Trees

The training of a RDF is conducted by training a number of trees independently. In order for the trees to offer independent predictions, training is performed as a stochastic process, decorrelating the trees. This is commonly achieved either by training each tree over a random partition of the labelled data (termed as bagging) or through random selection of the training features. Each Random Decision Tree consist of decision nodes which determine the traversal of a sample down the tree to the leaf nodes, which store the class distribution of previously seen examples in addition to any additional information recorded from training samples. Each of the decision nodes stores a splitting criterion ϕ and threshold θ which are learned during the tree growing process. Each decision node is considered a weak classifier which seeks to partition the training data $\mathcal{S}$ present at that node, into a left and right subset, $\mathcal{S}_l$ $\mathcal{S}_r$. The set with which a sample belongs is determined by,

$$\begin{aligned} \mathcal{S}_l(\phi) &= \{(\mathbf{I}, x) | F_\phi(\mathbf{I}, x) < \theta\} \\ \mathcal{S}_r(\phi) &= \mathcal{S} \setminus \mathcal{S}_l(\phi) \end{aligned} \tag{4.2}$$

where the parameters ϕ and θ are optimised on a per-node basis, and $\theta < \theta_d$, where θ_d is the depth of the hand. Learning begins at the root decision node, where a number of random candidate splitting criteria are evaluated. The performance of a candidate is determined by minimising an objective function.

The performance of each splitting criteria can be computed using several training objectives, which depend on the type of labelling present. For classification, the label entropy can be calculated, seeking well-defined classes at each split. The split which offers the largest gain in information is calculated using,

$$G(\phi) = H(S) - \sum_{p \in l,r} \frac{|\mathcal{S}_p(\phi)|}{|\mathcal{S}|} H(\mathcal{S}_p(\phi)), \tag{4.3}$$

where H is the Shannon entropy of the data labels. This term seeks splits that are informative while providing a balanced partition. This helps to maintain a globally balanced tree structure, which is favoured for run-time performance and to prevent overfitting.

An alternative objective would be to minimise the variance of continuous vectors. This allows the forest to perform regression with joint offsets stored in the leaf nodes. Tang proposed the use of multiple training objectives [115]. The first objective reduced pose variance (clustering global rotations), following by minimised label entropy and finally offset variance. Girshick [34] compared regression and classification objectives and found that the classification objective performed better for both classification and regression. The forests used in the remainder of this thesis utilise the classification objective due to the simplicity of a single objective function which can provide comparable performance for either regression or classification.

Each pixel of the hand provides a training datum consisting of its depth value d, at a position x in image $\mathbf{I}$ and a ground truth label c. The forest training process is computationally intensive, particularity due to the exhaustive assessing of the splitting criteria, therefore, sub-sampling must be performed. Pixel-wise random sampling is performed to ensure each of the poses in the dataset are represented. Spatial sampling is performed, sampling S_n from each pose example. While the sample distribution can be drawn uniformly across the labels, bias is favoured, due to the large variance in region sizes (fingertips, palm) which offers improved performance in validation.

Once the optimal split has been found for a node, the structure of the tree branches and creates two child nodes. The samples are then propagated to the newly created child nodes, based on the chosen split criteria. The training process continues, where each child node learns their own optimal splitting criteria, branching again. Training is performed depth-first, until a termination criterion is met, preventing overfitting. Learning terminates when either the trees reaches a maximum depth T_d, which restricts the number of nodes or when splitting offers a limited gain in information. Preventing over-fitting is an important aspect of training trees and is discussed in Section 4.2.2 which validates the depth used in training.

On branch termination, the final node is considered a leaf node. The leaf node records a normalised histogram of the sample-sets label distribution and provides the probability during prediction, which is aggregated across the whole forest.

On completion of training, out-of-bag re-evaluation of the label distributions is performed. This updates the histograms of the leaf nodes with unseen training examples [42]. All pixels from all hand images are propagated down the tree to replace the existing label distributions, forming new leaf distributions which reflect the complete data set.

4.1.3 Tree Classification

During evaluation, a test sample s propagates down the tree. The sample's path is determined by features stored in each node, which were learned during training. Each node evaluates the stored splitting criteria, passing the sample to either the left or right child node. Upon reaching a leaf node l, the probability of a sample's class $c' \in \mathcal{C}$ for the tree t is determined. Where $\mathcal{C}$ is the classes seen during training.

This is done for each tree independently and the probabilities are aggregated using,

$$P(c'|\mathbf{I}, x) = \frac{1}{|\mathcal{F}|} \sum_{t \in \mathcal{F}} P_t(c'|\mathbf{I}, x), \tag{4.4}$$

This results in the forest providing a more stable response than each tree alone.

The final class of the test sample is then determined as the label which has the maximal probability

$$L(x) = \underset{l \in 1..|\mathcal{C}|}{\arg\max}(P(c'|\mathbf{I}, x)). \tag{4.5}$$

Classification is completed for all pixels in $\mathbf{I}$ that belong to the hand, resulting in a bottom up region labelling. This means there is only an implicit dependency between neighbouring pixels. Several approaches have attempted to address this, both in the training process using Geodesic entanglement [46] and post classification using an MRF over superpixels [58]. However, their approaches lack kinematic dependency, which is revisited in Chapter 5.

4.1.4 Glove Based Labelling

In training a discriminative model such as a RDF, labelled data is required. This is challenging to acquire for the hand for a number of reasons. The large quantity of poses required and the number of joints makes manual annotation too costly. This is because a large number of images would need to be labelled with many three-dimensional landmarks. The positions must also be accurate and consistent, which is difficult due to the lack of textural information. Human observers can also find it difficult to differentiate ambiguous poses, leading to invalid labelling. Using a crowd-sourced approach will lack the accuracy and consistency required. While there are several datasets available that contain depth, a limited subset is annotated with joint positions.

Previous approaches have attempted to train solely using synthetic data, but their performance is heavily dependent on the realistic modelling of the hand [42] and the depth characteristics of the sensor [126]. The following section presents a revised framework that allows training using entirely real data that has scalable acquisition and is non-intrusive to the user or depth capture process. This results in simplifying the acquisition of large quantities of hand ground truth data.

Capturing using a Kinect camera provides calibrated depth and colour images of the hands. Employing a coloured glove provides the labels for training using real depth data. Augmentation using a glove offers reduced degradation of the depth image against active approaches allowing large quantities of real representative poses to be captured. However, the segmentation of the glove is an important aspect of its design. A glove design proposed by Wang [124] used geodesic triangulation, colouring the glove with 10 separate colours. A joint centric segmentation can also be applied that is based on Keskin's synthetic data. However, Wang's glove requires fewer colours, allowing improved colour separation and is used in the remainder of this chapter. Ideally the glove's colours must be determined empirically, choosing those colours for the glove that are most separable in the RGB camera of the Kinect. A preliminary study was conducted to account for the response of the Kinect's CCD. A short sequence was

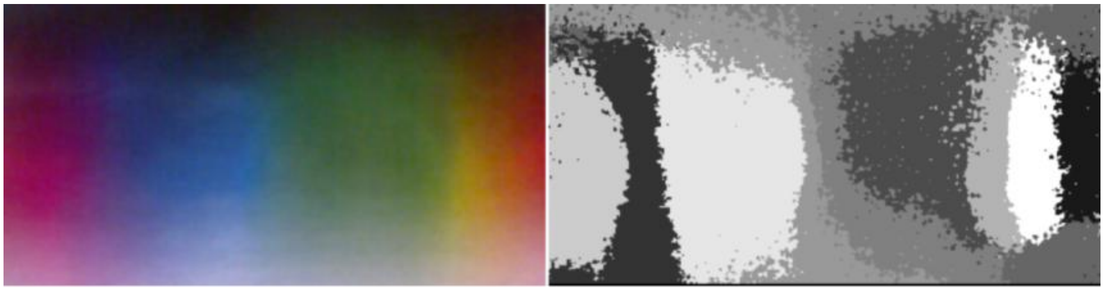


Figure 4.3: Isolated colour calibration chart as captured using the Kinect's colour sensor. Clustering outcome using k-means, labelling each of the colour pixels using the nearest cluster centroid.

captured using a colour calibration board, allowing the most separable colours to be found through clustering. Using k-means, 10 clusters were isolated in the LAB colour space. This was performed with the luma component removed to improve invariance against lighting conditions. The result of the clustering can be seen in Figure 4.3. Inspection of the clusters highlight that there are areas of the spectrum which are more separable, particularity greens, yellows, and reds, while blues form a single cluster.

The colours are then distributed across the glove such that the distance between regions with similar colours is maximised. The model was then rendered to a flat texture, which was printed on thin Lycra. The fabric was then sewn to form the tightly fitting glove used for performance capture. As can be seen, the presence of the glove does not affect the depth appearance (see Figure 4.4).

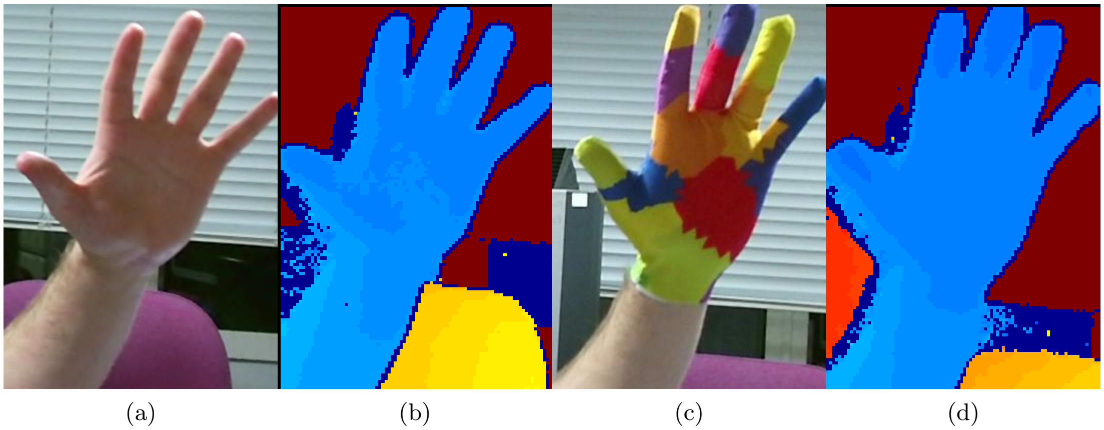


Figure 4.4: Comparison between the depth of a gloved and non-gloved hand. The depth images are very similar, with a slight edge dilation.

Each example consists of the calibrated depth image $\mathbf{D}$ and colour image $\mathbf{C}$. Using $\mathbf{D}(x)$ and $\mathbf{C}(x)$ the depth and colour of coordinate x is derived respectively. The label for each of the coloured pixels is then determined using a Gaussian Mixture Model (GMM) trained using several gloved examples that are manually annotated.

The GMM consists of a weighted combination of K Gaussian distributions allowing multi-modal data to be represented. Each colour of the glove is represented using an independent GMM labelled $l \in \mathbf{L}$,

$$P(l|\mathbf{C}(x)) = \sum_{i=1}^{K} \omega_i^l \mathcal{N}(\mathbf{C}(x); \mu_i^l, \sigma_i^l) \ , \tag{4.6}$$

where the weighting ω and Gaussian $\mathcal{N}$ with properties μ and σ are learnt from the training samples through expectation maximisation.

An additional background class l_0, with uniform probability, accommodates all other colours e.g. the background or glove boundaries.

This model can then be used to label large amounts of gloved hand images quickly and efficiently. Each pixel is compared against the model to determine the probability of it belonging to each class. The label that has the maximum likelihood is then chosen as per

$$l^* = \operatorname*{argmax}_{l \in L} \ P(l|\mathbf{C}(\mathbf{x})) \tag{4.7}$$

such that the depth sample is labelled l^*.

During experimentation, it was found that some colours of the glove are more susceptible to class confusion, despite modelling multiple modes. This was more typical of lighter colours such as green and yellow, due to specular highlights and shadowing which occurred during motion of the hand. Improved lighting constraints through the use of a ring light would likely reduce such issues. However, a relative likelihood is computed instead. This relative likelihood r is computed using;

$$r = \frac{P(l^*|\mathbf{C}(x))}{\max\limits_{l \in \mathcal{L} \setminus l^*} P(l|\mathbf{C}(x))} \tag{4.8}$$

This allows forest training to sample the hand using a weighted distribution, selecting pixels with easily distinguished colours. This is performed using a fitness proportionate selection, which is a probability based sampling method where samples with a higher ratio are selected first using the sorted values of r.

4.2 ICVL Evaluation

4.2.1 Dataset

The ICVL dataset was released by Tang [114] and was used to evaluate the performance of Latent Regression Forests. Latent Regression Forests are a regression-based approach which is used to localise the joint positions. This chapter focuses on applying the RDF to perform region segmentation. As such, comparison of joint error against Tang is discussed later, in Chapter 5. However, we can still use this dataset to evaluate training performance and validate parameter selection. The dataset is provided as depth images encoding mm depth from the camera. These images were captured using the Intel Senz 3D, which is a low-cost consumer time-of-flight sensor. The Senz 3D is capable of capturing 320 x 240 resolution images, and has a range of 0.15m to 1m. Over both sequences, the hand distance from the camera ranges from 0.24m to 0.42m. Hands in the range of 0.4m are approximately 100 pixels in width, demonstrating the limited resolution, despite the close range. The dataset covers poses ranging from pinching and grasping to rotation gestures. There are also rapid changes between poses and fast translations and rotations that test tracking robustness. Ground truth labels are acquired using automatic means, using the approach of Melax [67]. These landmarks were then manually corrected, with entirely erroneous frames being rejected. The landmarks consist of 16 3D part centres and have a depth that is internal to the surface. There are over 20000 labelled frames captured across 12 users which are synthetically rotated, providing over 300,000 training examples. There are two evaluation sequences of unseen users, each consisting of over 700 frames of labelled poses, again exhibiting challenging interaction poses and transitions.

Segmentation is not available. Labels are computed using a skinned hand model that is positioned using the ground truth landmarks. The depth image is then assigned indexed labels using corresponding regions on this model, which is discussed in greater detail in chapter 5. The resulting segmentation is shown in Figure 4.5. This is also applied to the NYU dataset.

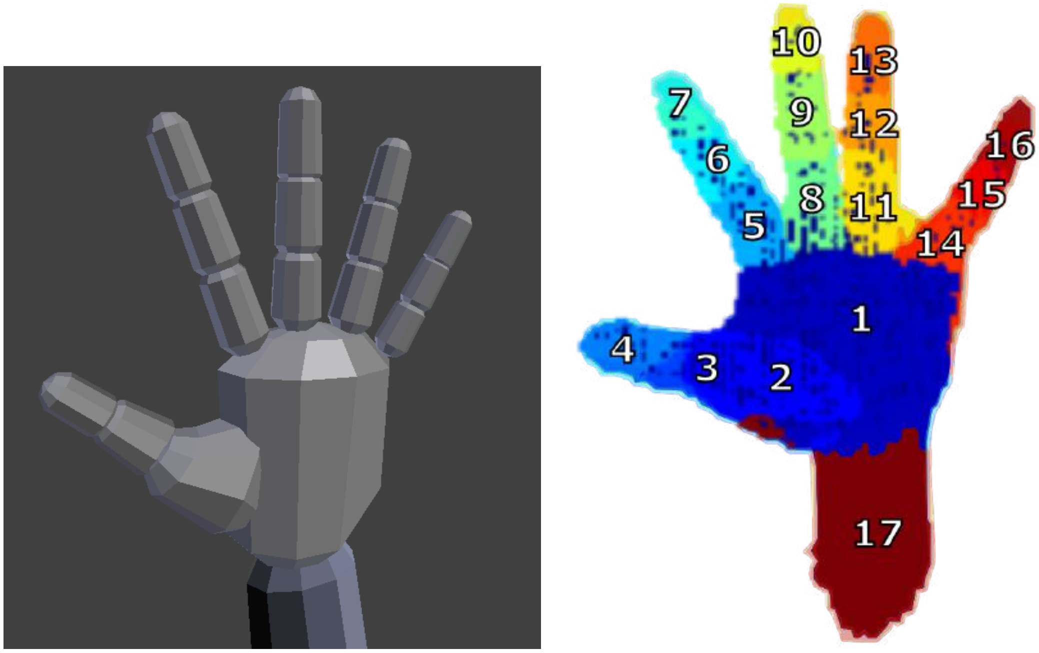


Figure 4.5: Hand Labels: Indexes assigned to each region of the model.

4.2.2 Parameter Selection

The forest has a number of tunable parameters, where optimisation can be performed against a validation set. Validation is performed using the ICVL dataset, due to the number of users available, one of which was excluded for validation, providing 8250 examples. This exclusion is important as the proportions and range of motion varies between users and we do not want to be biased to any particular user. The exploration of each feature is given below with quantitative measures of classification accuracy. The impact of training speed is also shown, as training a forest is computationally expensive.

Parameter selection used several default values that where replaced as more optimal values where found. These initial values used in parameters selection where chosen to provide good performance based on existing forest approaches in the literature, while favouring a reduced training time. Three trees were trained to a depth of 20, consistent with Keskin [42] and Shotton [98] and 1800 feature where evaluated, which was fewer than both [42] and [98]. This improved training time while providing a sufficient number of discriminative features. Finally the feature locality was set to half of the size of the average hand seen in the data, resulting in 50 pixels per meter.

Number of Trees. The following examines the impact of performance when altering the number of trees in the forest. Considering the tree as part of an ensemble, overfitting is reduced by training multiple trees using exclusive portions of the training set. Too many trees offer limited gain in performance, at the cost of increasing test complexity. The graph in Figure 4.6 shows the performance curve for an increasing number of trees.

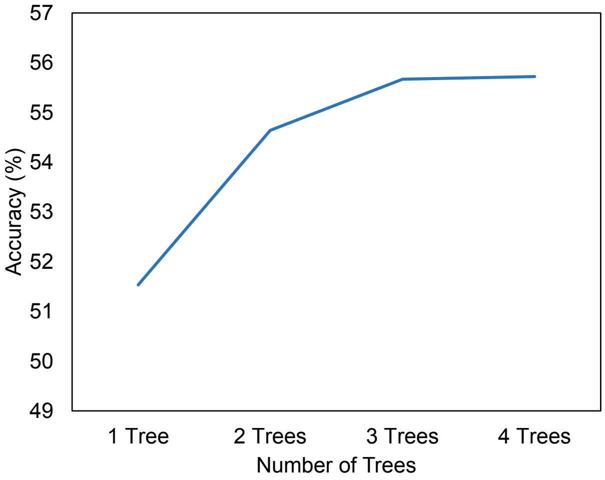


Figure 4.6: The classification performance in pixel-wise accuracy when increasing the number of trees in the forest. There is little gain in performance with the addition a fourth tree.

Given the marginal increase at four trees, three trees were used for the remainder of this thesis.

Number of Features. During training, the number of splitting criterion ϕ evaluated impacts the training time. However, increasing this number of features improves the likelihood of finding a good discriminative splitting criterion. The number of features explored were between 200 and 3400 features, in increments of 800 features. Figure 4.7 demonstrates the classification performance against the number of splitting criteria. The training time is also shown, demonstrating the increased training complexity.

It can be seen that there is a clear improvement when using more than 1000 features, but little gain over 2000. The number of features ϕ used in training was therefore set to 2000. N.B. this is consistent with both Keskin and Shottons findings.

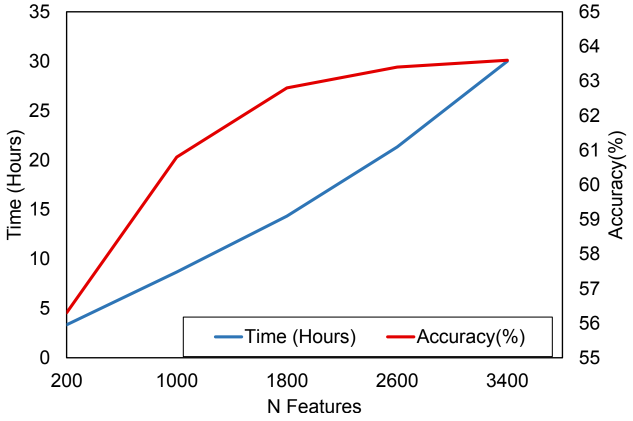


Figure 4.7: The classification accuracy with an increasing number of features evaluated at each tree node (ϕ). There is a consistent gain in training time with increasing the number of features. The improvement in performance reduces after 1800 features.

Tree Depth The tree depth can also be optimised by restricting the maximum depth of growth. By limiting the depth, the forests size is reduced which also prevents overfitting. The graph is shown in Figure 4.8 shows the change in performance with increasing depth. Performance can be seen to saturate once depth reaches 20. Beyond 25 the performance starts to drop as the trees overfit to the data. This is also consistent with both Keskin and Shotton. The trees used in the remainder of this thesis are trained to a maximal depth of 20.

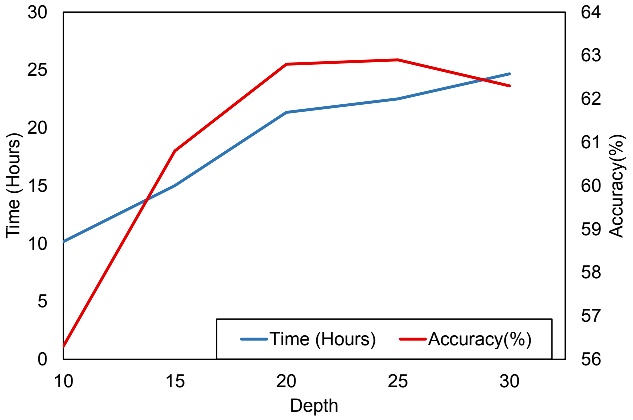


Figure 4.8: The classification performance of the forest compared against increasing depth. A depth greater than 20 reduces performance due to overfitting.

Feature locality. The offsets used in the splitting criteria sample a uniform distribution up to the maximum size of r_{max}. An increased radius will capture more of the hands context while a smaller radius increases density. It is expected that there will be a saturation in performance at the optimal radius. Again the performance is compared using the pixel-wise accuracy of classification. Figure 4.9 shows the increase in performance as the features reach 60 pixels per metre. The offset is quoted over metres as the offsets are normalised by each samples depth value. The curve suggests the optimal value is 60 pixels per metre which is used in the subsequent evaluation. This is to be expected as the hand is approximately 60 to 100 pixels across, at a metres distance.

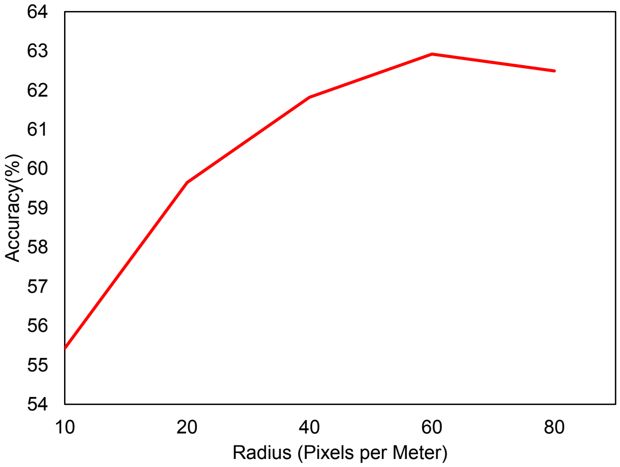


Figure 4.9: The classification performance when increasing the maximum radius of the offset features.

4.2.3 Results

The following section investigates the performance of the RDF using new and existing datasets. Ideally a comparison against previously published work would be conducted, however, a direct comparison cannot be performed against other approaches. This is because there are a limited number of approaches that perform a region based segmentation eg([71] and [42]). However, methods which perform segmentation have not published their training and evaluation datasets, making a direct comparison infeasible. In addition to this manual annotation of hand data is not viable, making ground truth data challenging to obtain. As such, approaches utilise synthetic training data. To allow comparison with approaches based on synthetic data, training data must also be provided as the quality of data is highly dependent on the hand model used to generate it. The forest was evaluated quantitatively by computing a percentage accuracy of correctly identified pixel labels. This measure is computed on each frame over the course of two challenging sequences, the results of which were plotted in Figure 4.10 and 4.11. Similar performance can be seen across both sequences with an average error of 50% in sequence 1 and 54% in sequence 2, which exhibits an increased range of poses. This performance can be explained with a closer inspection of the prediction results. Several examples labelled with ground truth segmentation can be seen in Figure 4.12(a) with the result of forest classification shown in 4.12(b). The classification error is shown in 4.12(c) where boundaries between classes shift. This can also be seen in the confusion matrix (Figure 4.13), where the highlighted structure is confusion between the base of the fingers the palm or between neighbouring fingers and joints (Figure 4.14). Such errors would lead a mode finding approach, used in conventional approaches, to incorrectly localise the joint, and motivates the contributions in Chapter 5. It can be seen that there is also large variance between successive frames which demonstrates the need for the incorporation of temporal information.

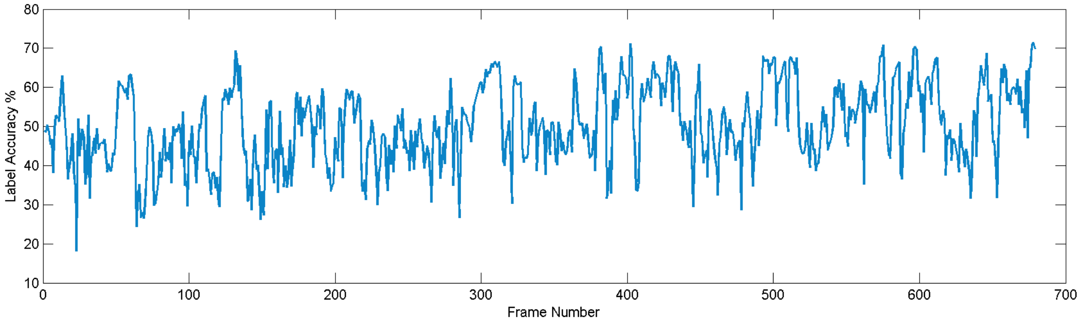


Figure 4.10: The classification accuracy for each frame in sequence 1 from the ICVL dataset.

4.3 NYU Evaluation

The NYU Dataset was published by Tompson [120] and demonstrates gesture based interaction. Such gestures include pinching twisting and grabbing.

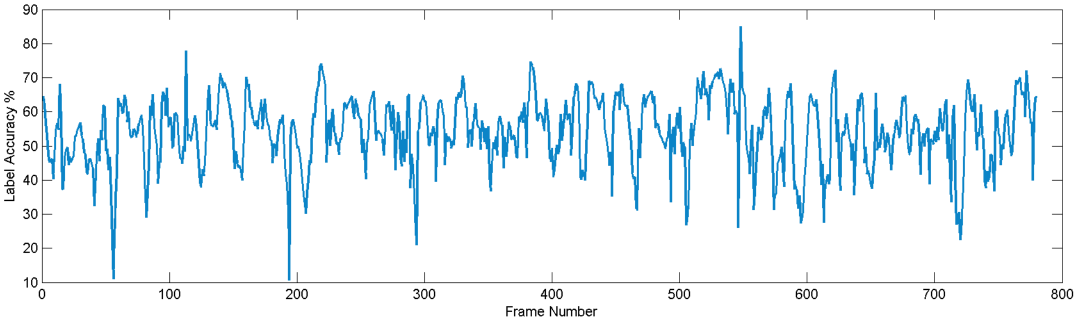


Figure 4.11: The classification accuracy for each frame in sequence 2 from the ICVL dataset.

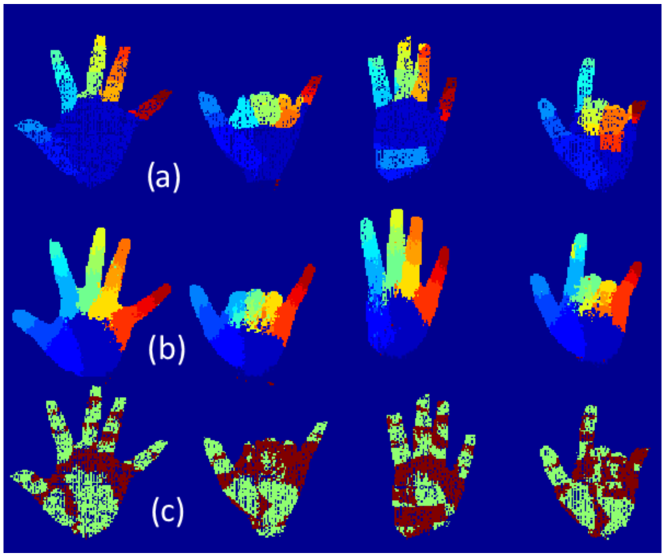


Figure 4.12: The ground truth of four example poses computed by fitting a model to the labelled joints (a). The resulting segmentation from forest classification (b). The incorrectly classified pixels are shown in red(c). (Labels are shown in Figure 4.5)

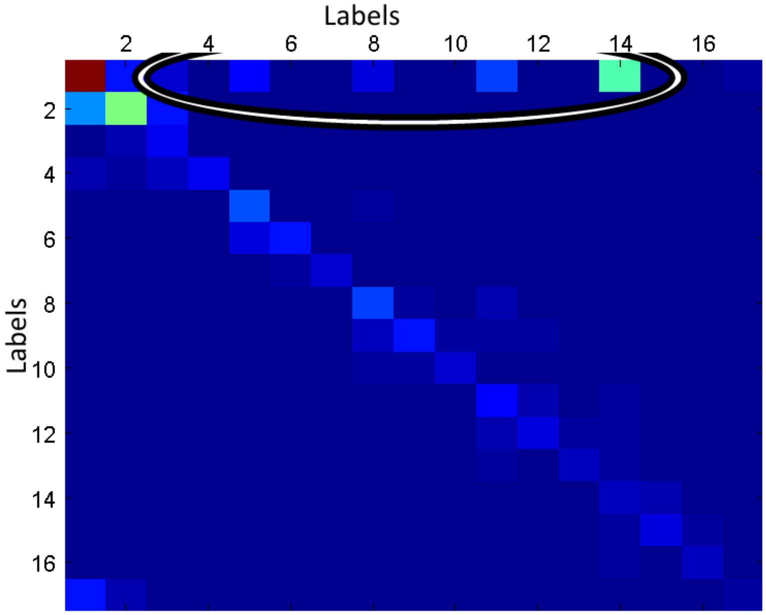


Figure 4.13: The confusion matrix computed from classification for all of the hand's regions. The confusion in the top row is between the base of each finger and palm. (Labels are shown in Figure 4.5)

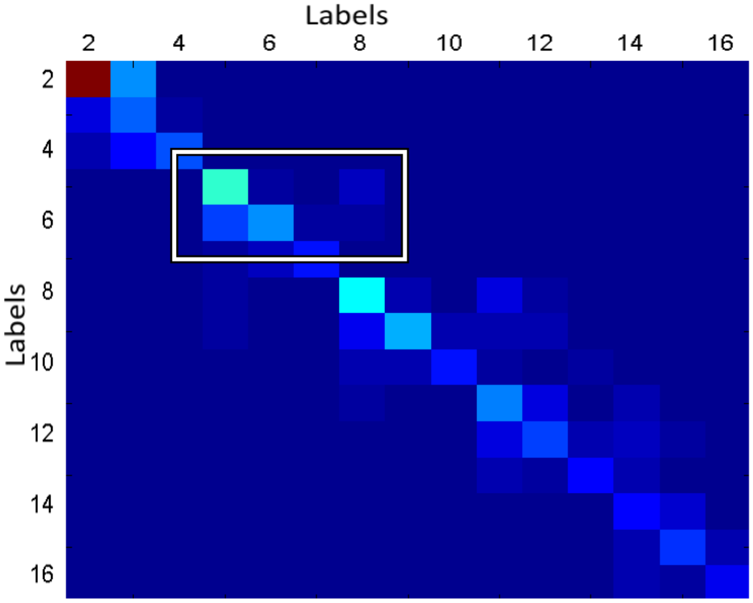


Figure 4.14: Rescaled confusion matrix of classification excluding the forearm and palm. A repeating structure can be seen in the confusion between neighbouring fingers and joints.

4.3.1 Dataset

The dataset was used to evaluate the performance of their CNN approach which performs direct regression of 14 landmarks. However, the example hands provided in the dataset are labelled with 36 landmarks. Segmentation was acquired through the model inference used in Section 4.2.1. The training set contains over 72000 training examples with 8000 test examples. The capture was performed using three Kinect cameras, with one positioned in front and two side views. Training contains only one user, limiting the variation while the test sequence has two uses. The dataset covers a range of motion with depths varying between 0.50m - 1.20m. As the data was captured with the first generation Kinect, the depth is subject to more noise than the ICVL dataset. In addition to this noise, a number of examples are missing fingers which point towards the depth sensor.

4.3.2 Results

Classification performance is demonstrated on the NYU dataset. A plot of the accuracy over the test sequences is shown in Figure 4.15, where an increased variance between frames can be observed. There is also a decrease in accuracy from frame 2441 onward. This is attributed to the sequence being comprised of two captures. The second sequence is of a different user not seen in training. The hand is also positioned with increased distance, with a depth of 1.2m compared to the initial depth of 0.77m. This makes it difficult to deduce if the reduction in performance is due to poor generalisation (one user in training) or the increased depth, which decreases resolution. NYU performs direct joint regression without providing segmentation. For this reason, direct comparison is presented in the following chapter, in which a method is proposed for the estimate of joints.

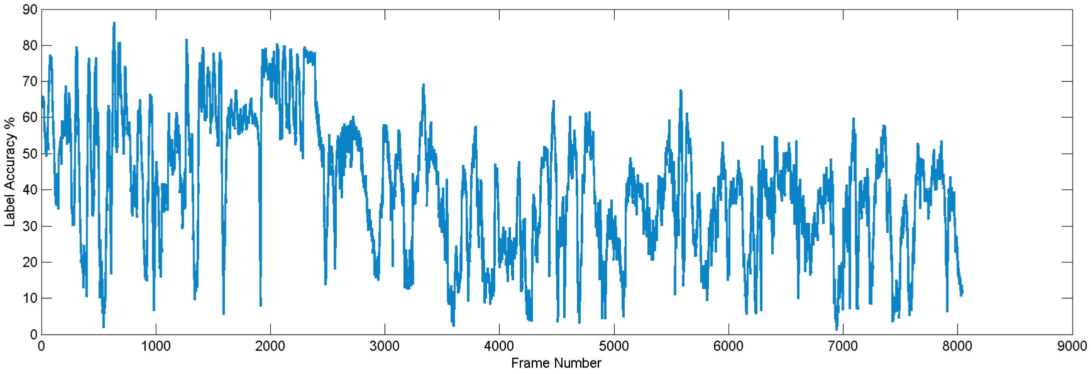


Figure 4.15: The classification accuracy for each frame in the test sequence from the NYU dataset.

4.4 Application to ASL

4.4.1 ASL Forest

To classify a hand example, each pixel of the hand is evaluated against the forest. Each sample traverses to a respective leaf node, with a prior letter distribution gained from training. The letter distribution from each pixel is then accumulated for each pixel in the image and the most likely letter returned. The approach proposed by Keskin required a two-stage learning process. The forest would provide segmentation of the hand which was used to localise the joints using a mode finding search. These locations then served as the features for a secondary SVM classifier. Instead we proposes a single learning solution. Inspired by the regression of joint offsets, the letter corresponding to the hand's shape is propagated down the tree. On arrival at a leaf node, the label is stored in a histogram with 24 bins, one for each character excluding J and Z, as they are dynamic variants. This exclusion is common practice for this task as additional temporal modelling is required for the full alphabet.

4.4.2 Dataset

Using the second-generation Kinect, high-resolution colour images were captured. Due to the size of the data, the hands were isolated using the Kinect's body skeleton tracking. Sequences were captured for four users performing ALS fingerspelling while wearing the

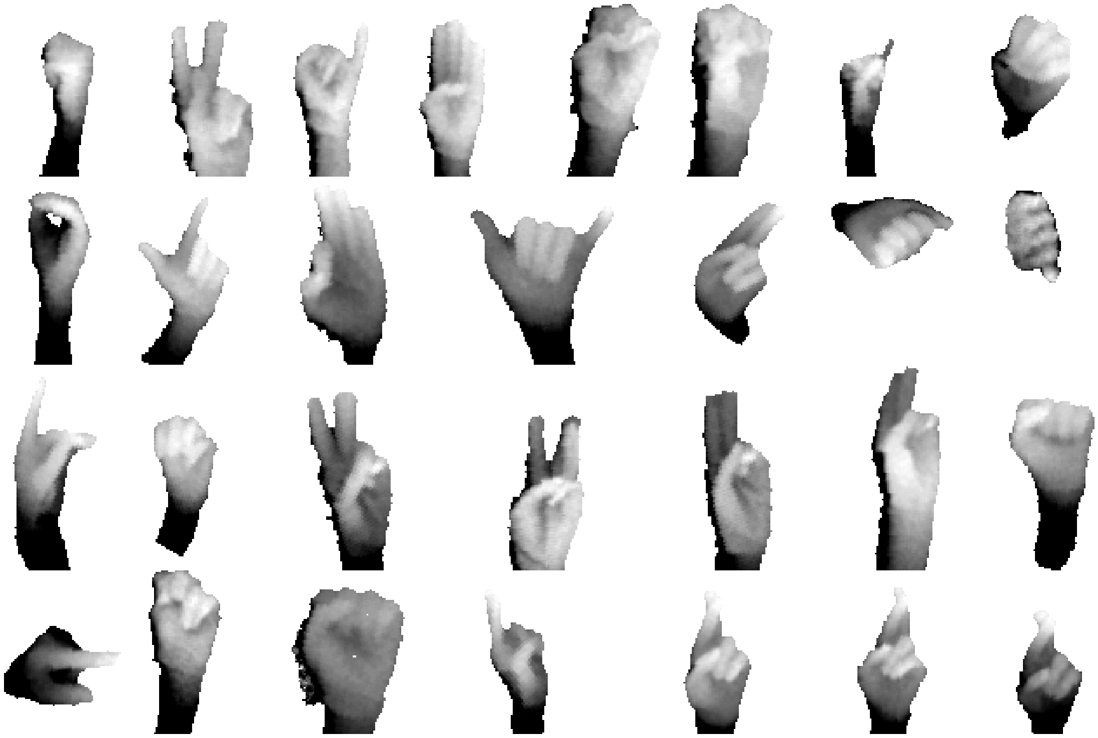


Figure 4.16: Examples from the ASL dataset, depth shows minimal impact from the glove.

lycra glove. The users found the glove comfortable, which was important for realistic capture. A training and test set were captured which were of equal size, with over 35000 examples in each. Each letter was represented by approximately 1500 examples, with varying vertical orientations. The dataset captures depth ranging from 0.60m to 1.20m from the Kinect. Examples present in the data are presented in Figure 4.16.

4.4.3 Results

Qualitative evaluation is provided for the result of colour segmentation derived from the GMM which can be seen in Figure 4.17. Pixel accuracy is also computed, and is shown in figure 4.18, where a large variance is shown between frames. The quantity of frames evaluated makes it challenging to see the error present. The quantity of frames evaluated makes it challenging to seen the error present. To show the accuracy of the classification clearly, a histogram is plotted against the classification accuracy. The histogram shows the majority of frames are classified with 80% accuracy. Those

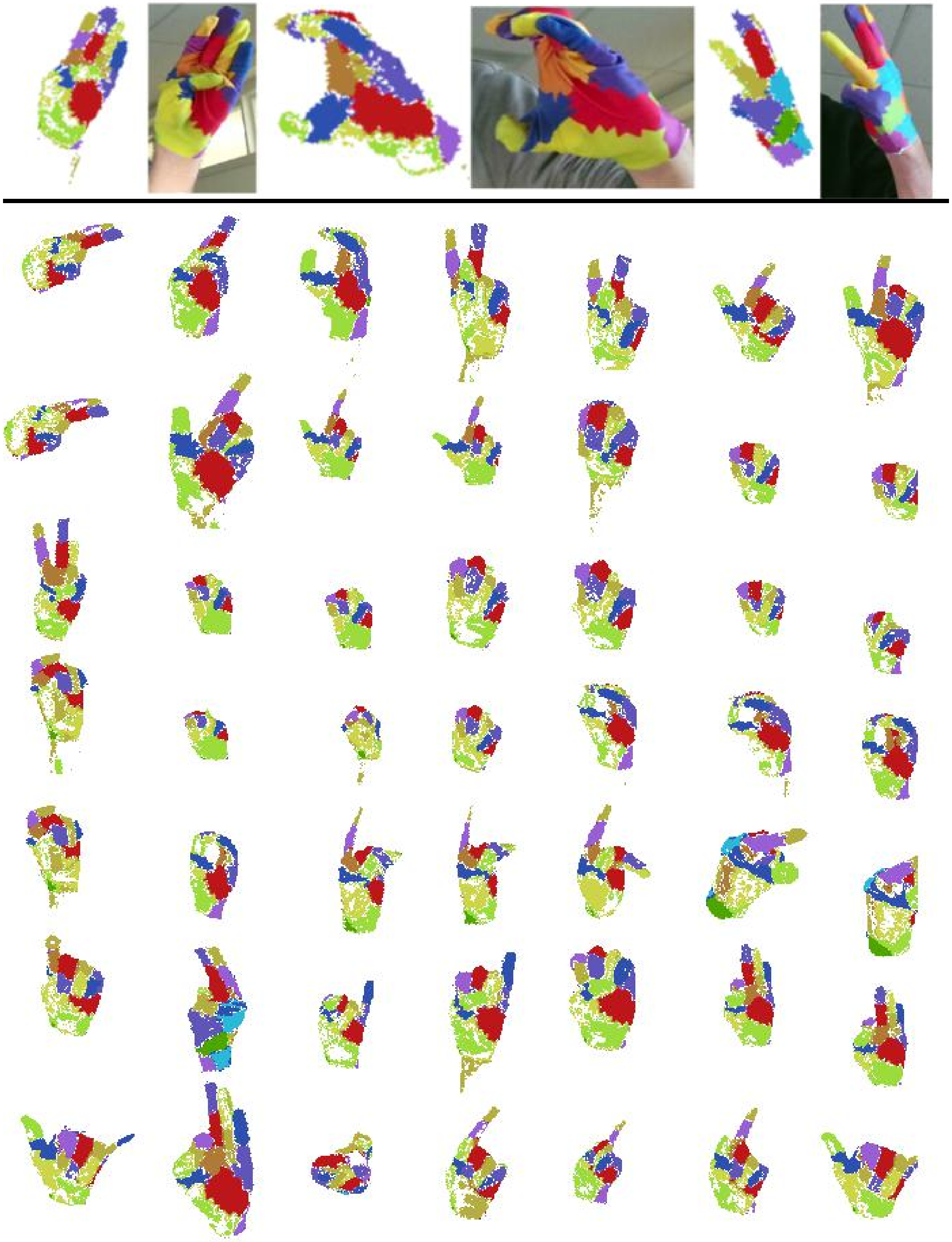


Figure 4.17: Examples of the GMM classifier with noise rejection used in forest training. Three examples show the input colour image.

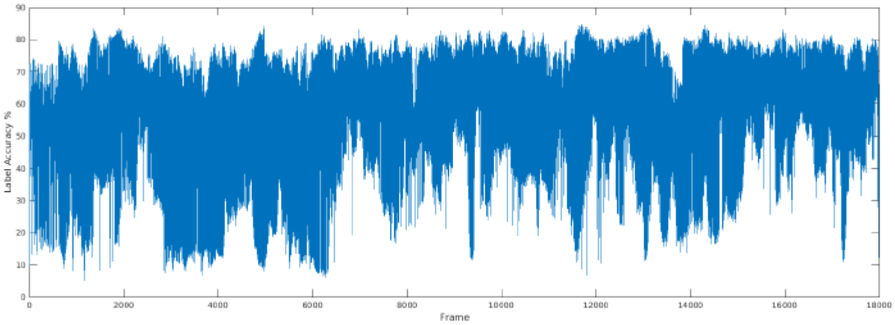


Figure 4.18: The pixel-wise classification accuracy for the ALS fingerspelling data, labelled using the glove.

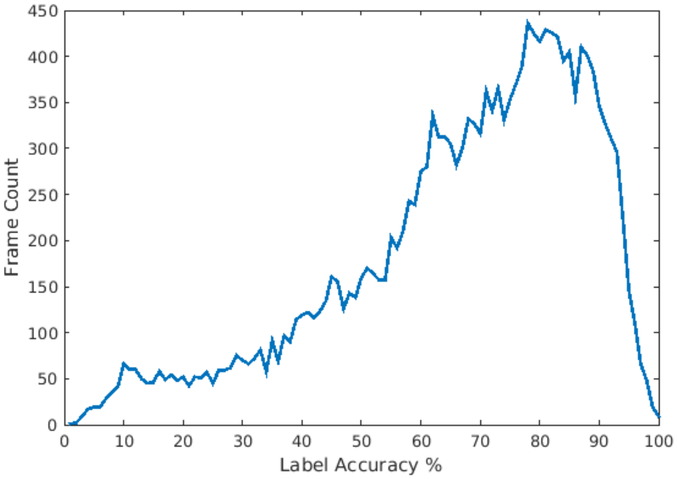


Figure 4.19: Histogram of classification accuracy shows the majority of frames being classified with 80% accuracy.

examples with less then approximately 30% accuracy are attributed to poses where the back of the hand are facing the camera. This is not typical of ASL but offer challenging examples to learn.

To evaluate the performance of the forest trained using a glove, classification of ASL fingerspelling is performed. Each hand is labelled with a predicted letter using the RDF and compared against the ground truth labels. The confusion matrix of this classification is shown in Figure 4.21. Again it is important to mention that J and Z are not evaluated. The accuracy of the labelling is computed by the trace of the confusion matrix giving 53% over 24 classes. It can be seen that there is a bias to the letters N with fist shaped poses and the letter R which also has similar poses. These poses can be seen highlighted in Figure 4.20. It not possible to directly compare against the performance of Keskin [42] as neither data nor code was made available.

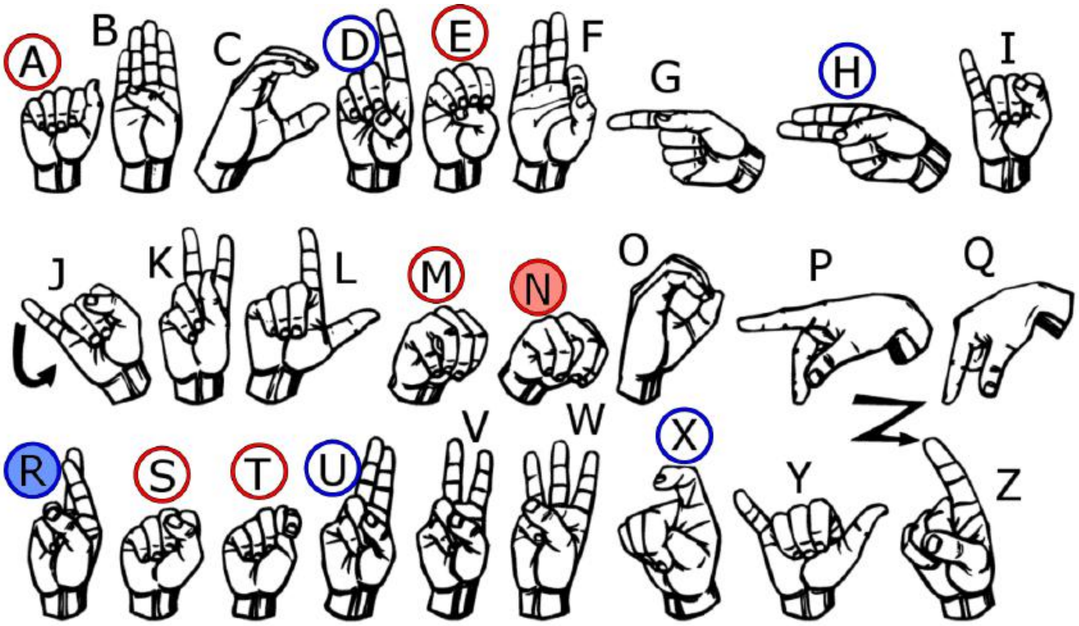
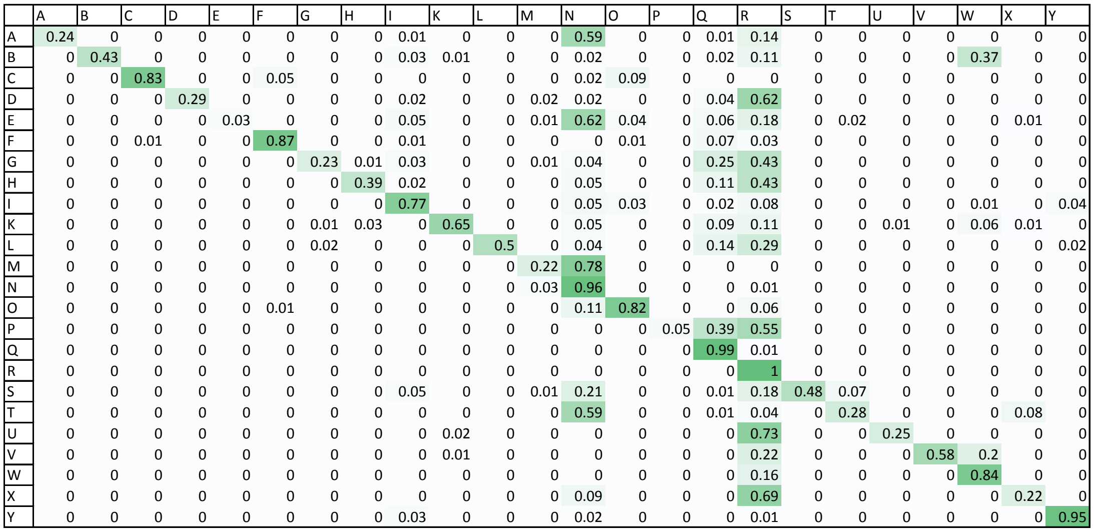


Figure 4.20: The ASL handshapes. The letters J and Z are dynamic, which requires additional learning. The letters N (Red) and R (Blue) have a number of similar poses, that makes them challenging to separate. Image source [60]



	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	0.24	0	0	0	0	0	0	0	0.01	0	0	0	0.59	0	0	0.01	0.14	0	0	0	0	0	0	0
B	0	0.43	0	0	0	0	0	0	0.03	0.01	0	0	0.02	0	0	0.02	0.11	0	0	0	0	0.37	0	0
C	0	0	0.83	0	0	0.05	0	0	0	0	0	0	0.02	0.09	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0.29	0	0	0	0	0.02	0	0	0.02	0.02	0	0	0.04	0.62	0	0	0	0	0	0	0
E	0	0	0	0	0.03	0	0	0	0.05	0	0	0.01	0.62	0.04	0	0.06	0.18	0	0.02	0	0	0	0.01	0
F	0	0	0.01	0	0	0.87	0	0	0.01	0	0	0	0	0.01	0	0.07	0.03	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0.23	0.01	0.03	0	0	0.01	0.04	0	0	0.25	0.43	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0.39	0.02	0	0	0	0.05	0	0	0.11	0.43	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0.77	0	0	0	0.05	0.03	0	0.02	0.08	0	0	0	0	0.01	0	0.04
K	0	0	0	0	0	0	0.01	0.03	0	0.65	0	0	0.05	0	0	0.09	0.11	0	0	0.01	0	0.06	0.01	0
L	0	0	0	0	0	0	0.02	0	0	0	0.5	0	0.04	0	0	0.14	0.29	0	0	0	0	0	0	0.02
M	0	0	0	0	0	0	0	0	0	0	0	0.22	0.78	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0.03	0.96	0	0	0	0.01	0	0	0	0	0	0	0
O	0	0	0	0	0	0.01	0	0	0	0	0	0	0.11	0.82	0	0	0.06	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0.39	0.55	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	0.01	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0.05	0	0	0.01	0.21	0	0	0.01	0.18	0.48	0.07	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0.59	0	0	0.01	0.04	0	0.28	0	0	0	0.08	0
U	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0.73	0	0	0.25	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0.22	0	0	0	0.58	0.2	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.16	0	0	0	0	0.84	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0.09	0	0	0	0.69	0	0	0	0	0	0.22	0
Y	0	0	0	0	0	0	0	0	0.03	0	0	0	0.02	0	0	0	0.01	0	0	0	0	0	0	0.95

Figure 4.21: The confusion matrix for classification of ASL fingerspelling, with autonomously labelled training data.

4.5 Conclusions

In summary, this chapter presented Randomised Decision Forests (RDFs), and their application to hand pose estimation. The forest is a machine learning approach capable of learning a range of hand poses which were trained using thousands of examples. The forest discussed in this work was trained to perform classification, allowing the regions of the hand to be labelled. Forests are also fast to evaluate, segmenting the hand in real time from a single depth image. Trained using large datasets, the forest learns discriminative features. These features represent simple structures, computed using pair-wise measurements of depth. Alone these features offer limited discriminative power, but as seen in the evaluation, complex structures can be modelled. This is attributed to the structure of each tree in the forest. Additionally, the use of multiple trees provides improved generalisation. Information can also be propagated down the trees structure, as seen by the application for ASL fingerspelling recognition.

This chapter also introduced the use of a coloured glove as a training aid for automatic ground truth labelling. Using both modalities of colour and depth, automated annotation could be performed. Using a GMM classifier trained using several labelled examples facilitated the labelling of thousands of examples. The GMM was formalised such that multiple colour modes can be modelled, distinguishing between 10 coloured regions. To improve robustness to varying lighting, the model also incorporated a relative measure to reduce sampling from ambiguous labels.

The approach was quantitatively and qualitatively evaluated on challenging datasets with varied applications. Excellent performance was exhibited on ASL finger-spelling hand shapes, but further investigation found a failure mode in performance when classifying pixels, due to the boundaries of regions. Frame wise analysis also showed variance between successive frames, which demonstrates the need to incorporate temporal information. It was observed that several similar hand shapes are difficult to discriminate, which reduces performance. A possible solution would be to recolour the glove such that regions important for distinguishing similar poses could be labelled with more strongly contrasting colours. The subsequent chapter looks to introduce temporal information.

Chapter 5

Combined Decent Optimisation

5.1 Introduction

This chapter presents an approach to hand pose estimation that combines discriminative, model-based and regression methods to leverage the advantages of each. The segmentation from the previous chapter forms the basis of constraints applied in model fitting, which enforces temporal continuity and kinematic limitations. The approach provides improved accuracy over the current state of the art methods, through the inclusion of temporal cohesion and by learning how to correct failure cases. The combination of region labelling and constraint-driven optimisation allows tracking to be performed at 40 frames per second using a single CPU thread.

Existing discriminative approaches utilise large datasets to capture the variety of poses [43,114,120]. Often, these approaches evaluate the part positions using a single frame . However, this discards prior information and temporal cohesion, which can be used to reduce the likely pose space and eliminate jitter. Alternatively, model-based approaches [67,72] depend on temporal information but can become trapped in local minima in the optimisation process which requires reinitialisation. This chapter will demonstrate that combining both approaches reduces these mutually exclusive failure cases, improving robustness and accuracy.

We use classification of hand regions to reduce the pose space and guide a fast model-based optimisation. This allows the explicit modelling of kinematic constraints while preventing self-intersection, and provides smooth realistic tracking of the hand's parts. Reinitialisation is also handled implicitly, as optimisation is guided at each step from the results of segmentation. Error correction is then learned from failure cases and the residual error is used to refine estimates and overcome user variation, providing state of the art performance on benchmark data.

The performance stems from 3 main contributions. First, the Constraint Driven Optimisation (CDO) used for model fitting, which uses a combination of model optimisation and discriminative methods, incorporating prior knowledge into model optimisation. Secondly, improved accuracy with the introduction of cascaded linear regression, which corrects the residual error. Finally, direct sampling of residual error for training, which captures system response and user variance, allowing supervised feedback for part refinement.

5.2 Method Overview

Given a parametric representation, estimating the pose of the hand can be conducted through optimisation, finding the parameters of a hand model that minimises the error between the observation and the models appearance. In the case of optimising against depth, the models appearance is typically rendered as a depth map, which is then compared against the observation. Optimising a hand model using depth is challenging due to the hand's high DoF. Its complex structure and range of local deformation, as well as global transformations, mean that many different pose configurations can have a similar appearance, leading to local minima that break optimisation, and require reinitialisation.

The proposed approach differs in that optimisation is heavily constrained using discriminative segmentation of the hand. A hand model, constructed using 3D bodies was designed to imitate both the shape and limits of real hands. Then, using a Newtonian based simulation, the dynamic behaviour of the model is synthesised, such that forces applied to the model determine its position and orientation. The pose of the model is then determined by the attachment of the depth observation (Figure 5.1c) through

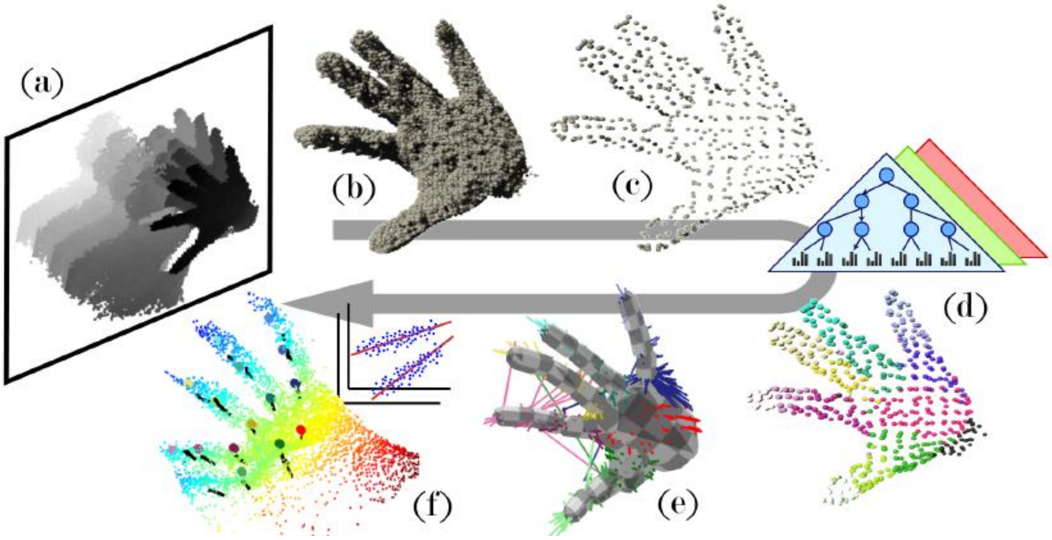


Figure 5.1: Method overview which shows processing of the depth camera stream. (a,b,c) The depth is converted to its cosponsoring point cloud for filtering and sub-sampling. (d) Forest classification labels each point using depth. (e) This provides correspondence for constraint-driven optimisation. (f) Linear regression using depth sampled features then corrects model discrepancies.

spring based, point to surface constraints. These constraints pull the model into position acting similarly to ICP, which minimises the error between model and observation. This heavily constrained system of bodies is resolved using a Projected Gauss-Seidel solver. Through the iterative application of impulse forces, the model's position moves closer to the observation (Figure 5.1e). Unique to our approach, point correspondences for each constraint are determined using a Randomised Decision Forest seen in Figure 5.1d. This allows the incorporation of previously seen examples, using a technique that is fast to compute. The progressive update of model position based on Newtonian dynamics incorporates the temporal information, ensuring realistic dynamic behaviour and reduces inter-frame jitter. For these reasons, the combination of both approaches improves the performance of either alone.

There are two sources of error inherent to this approach. Firstly, the use of a general model for varying users without re-targeting will result in a residual error, which limits accuracy. Secondly, as the model fitting is gradient descent and subject to local minima or false minima due to errors in segmentation. There for the final part positions are refined through cascaded linear regression, seen in Figure 5.1f. The result of the earlier model-based optimisation serves as the initial estimate. The cascade itself operates using several tiers which provides an approximate piecewise linear regression solution to what is actually a high-dimensional non-linear problem. High dimensional features are sampled from the prior stage capturing local context around each part. These features are projected through each tier's linear model, iteratively refining the error between the estimate and the correct part location. This latter stage accounts for hand-model variances and improves accuracy, learning from failure cases previously seen in model optimisation during training.

5.3 Hand Pose Estimation Approach

The following section details the approaches used at each stage of the hand pose estimation framework.

5.3.1 Segmentation and Filtering

The hand is first segmented from the image using depth. For the purposes of evaluation, the hand is assumed to be the closest object to the camera. Alternative methods capable of tracking the users skeleton would allow more robust separation, but are outside the scope of this work. The points of the hand are labelled as $\mathcal{P}$, defined in Section 3.1. Unlike Chapter 3 the points of the wrist are included in the subsequent stages as the RDF is trained to label the forearm which provides improved segmentation. As in chapter 4, the depth of a pixel $\mathbf{x}$ in image $\mathbf{I}$ is accessed using $d_I(\mathbf{x})$.

The set of points $\mathcal{P}$ is very dense as hands captured by short-range depth cameras produce upward of 5000 cloud points. The spatial resolution is much higher than necessary for model optimisation. Camera noise around the contour of the hand can also be observed in the cloud and these erroneous points are likely to impact accuracy. Intelligently down sampling the cloud to representative points reduces complexity, noise and improves performance of the subsequent stages. It is common practice for model based optimisation [67,85]. To achieve this, the application of a Voxel grid filter provides both downsampling and outlier rejection and can be seen in Figure 5.2. Voxels with

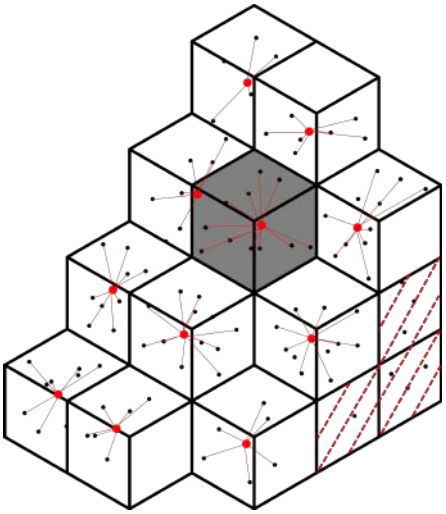


Figure 5.2: Example of voxel filtering, down sampling a point cloud (black), to its resulting centroids (red). Hatched voxels represent those points (outliers) rejected due to insufficient support.

an insufficient number of points are removed, signified by the red hatches, rejecting outlier noise that forms around the contour. The points (drawn in black) within the remaining voxels are then down sampled, reducing the collection of points to their centroids shown in red. This process provides $\mathcal{P}' \subset \mathcal{P}$, a representation of the original cloud that maintains detail while significantly reducing density. The strength of filtering can be tuned by changing the voxel size. The larger the voxel size, the less dense the point cloud, and vice versa. Consideration in choosing voxel size must be given to ensure that small features of the hand (such as the fingertips) are represented. The voxels are 2.5mm in size, providing sufficient detail of the fingers . The filtered point cloud is then used during optimisation in the following section.

5.3.2 Kinematic Hand Model

The hand configuration is estimated through optimisation of a generic hand model against the filtered point cloud. Ideally a model must have similar shape and proportions to the real hand it is optimised against and be able to synthesise its dynamic behaviour, which includes realistic representation of joint flexibility. Kinematic constraints enforce the fact that only viable poses are generated, reducing the optimisation search space.

During flexion of the joints, the surface geometry of the hand changes. The inclusion of muscle and bone meshes would add realistic bulging and sliding of the skin [1]. However, the model optimisation omits such complexity as the Residual Error Regressor (RER) in Section 5.4 aims to resolve surface variance.

Changes in global scale can be adjusted at runtime [67], however, changing the proportions of the hand to match a user is more challenging and extends the DoF of the model [62]. User specific models could be constructed [117], but would require a calibration stage. To avoid this, the approach optimises a generic model and refines the pose against such variance through RER.

The use of a mean hand allows tractable optimisation that generalises across users. Morphological surveys of the human hand [15] measure such variance. Using this information and multiple reference images across several users, a general hand model was constructed. The hand model $\mathcal{H} = b_1 \cup b_2 \cup ... \; b_n$ is comprised of $n = 17$ bodies shown in Figure 5.3a, three capsules per finger and thumb while a single body models the palm and wrist. This use of a single mesh for the palm is suitable for optimisation but there is limited flexibility across the metacarpals. Using convex shapes not only simplifies model construction but also allows point to surface correspondences to be calculated quickly, discussed in Section 5.3.4. Each of the bodies are connected through a skeletal hierarchy rooted at the wrist using hinge and rotational constraints to reflect the anatomical structure. The skeletal structure can be seen in Figure 5.3b and can be attached to a weighted model for realistic animation. These kinematic limitations are applied with ranges that match those proposed in [61]. For the purpose of segmenting training data, the forearm is also modelled with realistic kinematics.

By performing the optimisation in a Rigid Body Simulation (RBS) framework, temporal tracking and prediction is implicitly modelled. The mass of each hand component impacts the acceleration during the application of constraining forces. Assuming a constant density of the hand the mass of each component is estimated using the volume of each convex shape. This allows the motion of the palm to have greater kinetic influence over the fingers, allowing realistic tracking with rapid global motion.

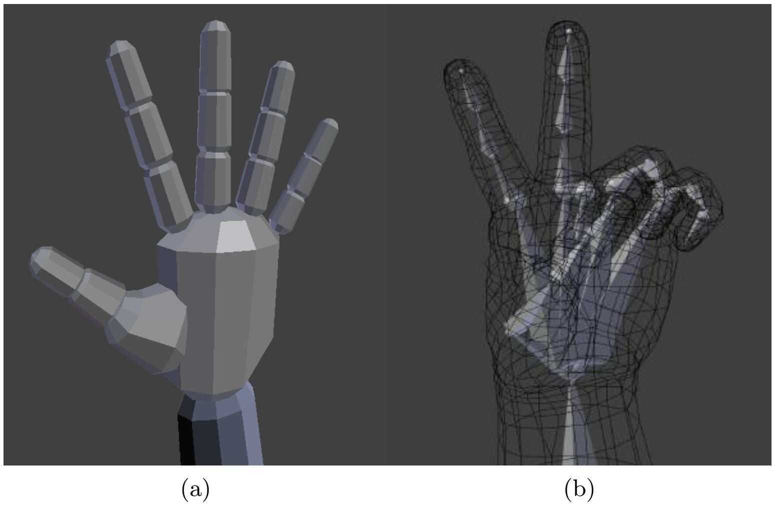


Figure 5.3: Kinematic hand model constructed in Maya. (a) The hand is comprised of separate rigid bodies, joined through constraints to reflect realistic limitations. (b) The model is also rigged with a poseable skeleton allowing the generation of training data, and use in 3D applications.

5.3.3 Rigid Body Simulation

By applying RBS to our kinematic model, we aim to identify the pose configuration that minimises the error against the filtered point cloud. RBS is used in the modelling of physics driven interactions between objects, most commonly used in film and games [14]. It is comprised of a number of algorithms that aim to resolve the position and forces applied to bodies in the simulation. Object collision is examined through a two stage process with the aim of preventing self-intersection of the fingers. A broad-phase eliminates those bodies too distant to collide, while a narrow-phase confirms and localises the point of contact between colliding bodies. It is important to note that the simulation takes place in discrete time steps meaning colliding objects intersect. This intersection is computed efficiently for convex bodies using the Gilbert-Johnson-Keerthi distance algorithm (GJK) [33]. On collision, a repulsive pairwise constraint is applied pushing the bodies apart.

System constraints are resolved using a Projected Gauss-Seidel solver which is formulated to reproduce realistic Newtonian physics. This derivation from Newtons laws of motion enforces temporal cohesion and each bodys motion state is modelled. Constraints are enforced through the application of impulse forces on a pairwise basis, the direction and magnitude of which, aim to reduce the constraint error. Several iterations are performed over each time step, minimising the error between successive frames. This jointly enforces the kinematic limitations of the joints and the collision constraints applied in the previous stage.

5.3.4 Point to Surface Constraints

Point to surface constraints ensure contact between a point **p** and the surface **s** of a rigid body $\mathbf{b_i}$. The constraint solver determines the impulse forces needed to minimize the constraint error. This error is calculated as the residual between **p** and $\mathbf{p}'$, where $\mathbf{p}'$ is the closest surface point.

$$\mathbf{p}' = \underset{\mathbf{p}_b \in b_i}{\arg\min}(||\mathbf{p} - \mathbf{p}_b||) \tag{5.1}$$

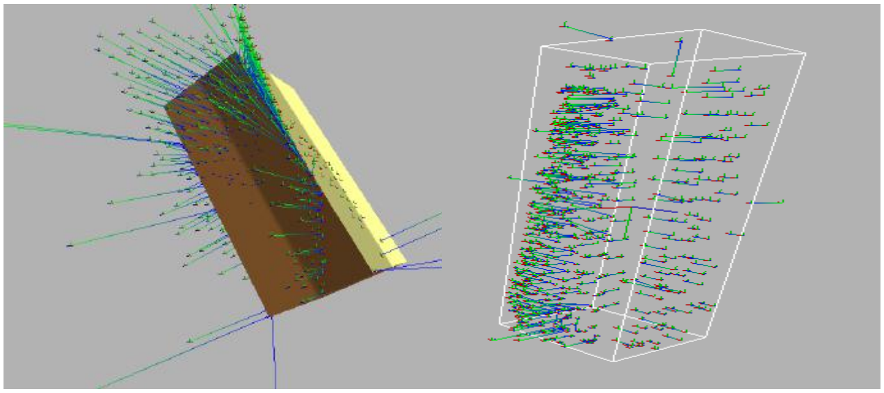

(a) (b)

Figure 5.4: Using point to surface constraints to bind a single cuboid to the point cloud

The closest point needs to be updated during the simulation to reflect changes in position and orientation of the constrained body. This again would be computationally intensive to perform without the use of the GJK distance. GJK is an optimised method to determine intersection/closest points between two convex polygonal objects. The approach was modified to find the closest point on a body b_i to a point **p** in 3D space, which serves as the attachment point for the constraint. This location updates at each iteration, acting as a point to surface constraint. The application of many point to surface constraints is shown in Figure 5.4 illustrating the optimisation of fitting a cuboid to its point cloud. Figure 5.4a shows the attachment of constraints to their closest point on the cuboid surface prior to optimisation, while Figure 5.4b shows the resulting pose along with the updated surface constraints.

The point cloud captured from a depth sensor represents the visible surface of the hand, as such, a sample $\mathbf{p}$ in the point cloud must correspond to a position on the camera facing side of the hand. It is challenging to determine where on the hand this point resides, due to the lack of textural information. An exhaustive search for a model configuration that satisfies these constraints is not tractable. Searching locally to a prior estimate reduces the search, but during rapid motion, large transitions between frames lead to local minima. Such errors are difficult to detect and require reinitialisation. Instead, we estimate correspondence to the model using the Randomised Decision Forest (RDF) which utilises global spatial context. This Constraint Driven Optimisation (CDO) allows the incorporation of a priori knowledge from training when optimising the hand model.

5.3.5 RDF Assignment

The RDF learns discriminative features which allow the region labels to be determined but the training of the forest requires segmented hand images which are provided during training. These regions were initially found using the nearest neighbour assignment of each of the hand's pixels to its nearest part in 3d space. However, this was found to be unreliable as the assignment did not consider the boundaries between fingers and lacked an understanding of occlusion. This can be seen in Figure 5.5a with noise present around each of fingers.

A better solution was to use the hand model $\mathcal{H}$ to label the pixels as belonging to each part of the hand. For each example pose, the model $\mathcal{H}$ pose was estimated through optimisation to match the ground truth, constraining each part's position to its corresponding labelled landmark. Each point in the depth image was then back projected

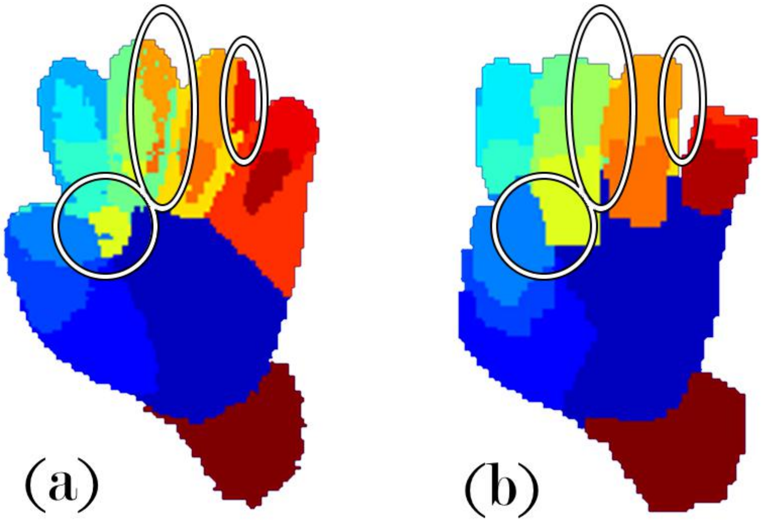


Figure 5.5: Figure demonstrating labelling of the hand using ground truth landmarks. (a) An example pose labelled by assigning each pixel the index of its closest part. (b) Using a ray test through the model, which is constrained using the ground truth

using the cameras intrinsic calibration **K** to its 3D position. A ray was then traced from the camera to each point and beyond, thus the correct hand part could be identified at the intersection between ray and hand model. This provided robust labelling and offers two distinct benefits; firstly the segmentation is occlusion aware, recognising closely interacting fingers, and secondly the segmentation is accurate to the model, with consistent labelling at the boundary of neighbouring parts. This improved method of label assignment can be seen in Figure 5.5b under each of the ellipses. In addition to labelling the hand, the forearm was also identified to improve the localisation of the palm when δ_d is overestimated.

At runtime, each point $\mathbf{p}$ in the point cloud $\mathcal{P}'$ is projected on to $\mathbf{I}$. The point is then assigned a label computed by the forest classification, corresponding to a part b_i.

Once each point's label is found, point to surface constraints are assigned to their appropriate body. The RBS is then solved, using a projected Gauss-Seidel solver. Optimisation iteratively applies impulse forces to reduce the error in equation 5.2 from

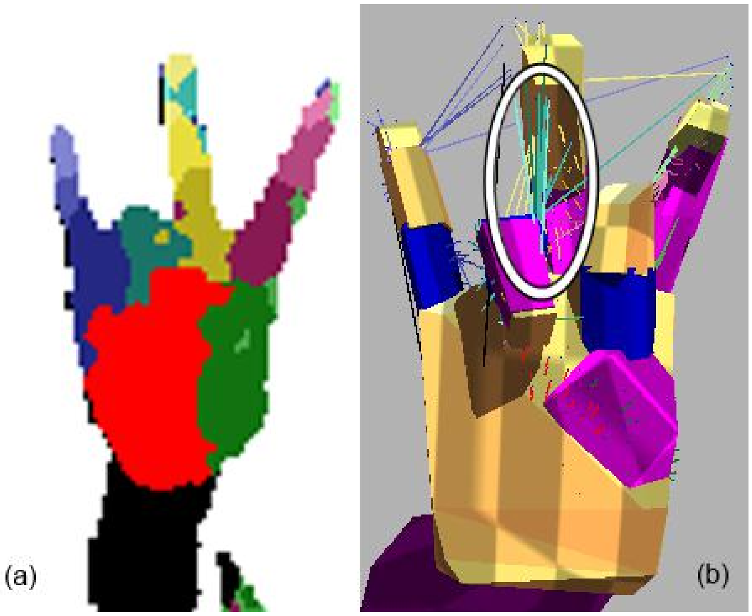


Figure 5.6: Constraint driven optimisation. (a) The hand's depth following segmentation using the Random Decision Forest. (b) The kinematic hand model after optimization using constraints. The ellipse highlights the finger being correctly estimated, despite erroneous constraints that are overpowered through optimisation.

all point to surface constraints and kinematic constraints.

$$\mathbf{p}' = \arg\min_{\mathbf{p}_b \in b_i}(||\mathbf{p} - \mathbf{p}_b||), \text{where } i = L(\mathbf{p}) \tag{5.2}$$

and $L(\mathbf{p})$ is defined in Equation 4.7. Constraints which are incorrect are overpowered by the global consensus. This can be seen in Figure 5.6 where the middle fingertip is correctly identified, shown as yellow in 5.6a, which pulls the model correctly into position, despite the misclassification of the middle section. Had the part centre been determined using mean-shift or regression, it is likely the resulting fingers would self-intersect, providing an invalid pose estimate.

5.4 Residual Error Regression

This section discusses our Residual Error Regressor (RER) which aims to learn to recover from failures in optimisation due to local minima and residual errors in model fitting. One source of residual error comes from discrepancies between the user's hand

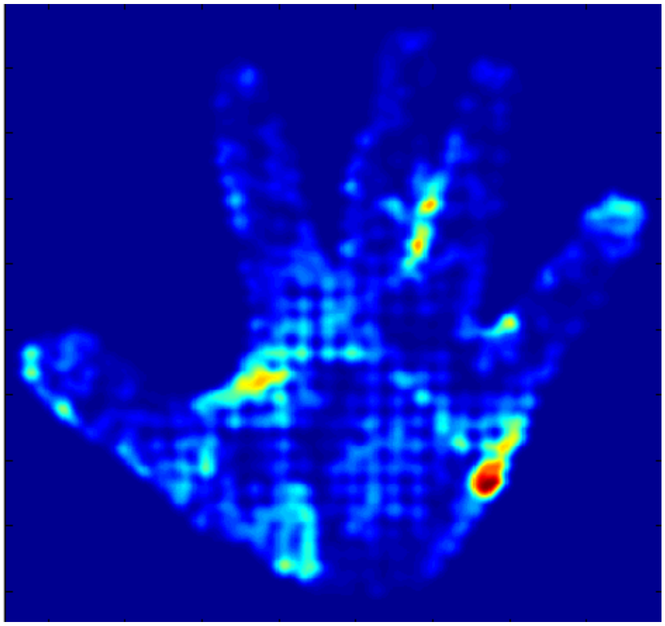


Figure 5.7: Residual error in using a generic model against a user with different proportions. This error is attributed to the use of a general hand model, and inter-person variation.

shape and that of a generic model. This residual error can be seen in Figure 5.7, which shows poor fitting around the palm and some parts of the fingers. The additional optimisation of hand proportions can be resolved in real-time using a model with increased degrees of freedom [62]. Instead, we propose the use of discriminative means to provide correction offsets that refine the pose estimate. Using local features around each estimated part, a cascaded linear regression is trained. Training samples are generated using pose estimates, computed using the Constraint Driven Optimisation (CDO) discussed in Section 5.3.4 with perturbed initialisation.

The pose estimated via CDO serves as the initial estimate and is defined as $\mathcal{H}^0$. The centroid of each body is defined as $\mathbf{c}_b$ during refinement. Information regarding the surface gradient and contour is captured using features that use pairwise offset $\mathbf{u}$ and $\mathbf{v}$ derived from 4.1 given as follows:

$$F_{\mathbf{u},\mathbf{v}}\left(\mathbf{I}, \mathbf{c}_b\right) = d_I\left(\mathbf{x}_b + \frac{\mathbf{u}}{z}\right) - d_I\left(\mathbf{x}_b + \frac{\mathbf{v}}{z}\right) \tag{5.3}$$

where z is the depth of the part centroid $\mathbf{c}_b$ rather than image depth and $\mathbf{x}_b$ is its projected image point. Features are sampled around each part $\mathbf{c}_b \in \mathcal{H}$, providing local context about the depth surrounding its location. f features are sampled randomly around each part, with the difference in depth between offsets being constrained to the range of the hand's depth.

The features for the hand's parts are then concatenated in part order (Figure 4.5) forming a high dimensional feature vector representing $\mathcal{H}$. Due to the sparse nature of the features, PCA is performed. Those dimensions containing 95% variance over the training set are preserved. We define the function $\phi(\mathbf{I}, \mathcal{H})$ to denote the PCA projected features.

Each tier of the cascade $k = (1, ..., K)$ estimates a part's update in pose $\Delta\mathcal{H}^k$, which aims to converge to the true part's positions. The following describes the application of the offset vector computed by each independent regressor $R(\phi)$

$$\begin{aligned} \Delta\mathcal{H}^k &= R^k(\phi(\mathbf{I}, \mathcal{H}^{k-1})) \\ \mathcal{H}^k &= \mathcal{H}^{k-1} + \Delta\mathcal{H}^k \end{aligned} \tag{5.4}$$

The pose refinement $\Delta\mathcal{H}^k$ predicted by $R(\phi)$ uses a linear projection of the high dimensional features ϕ. Training for each tier of regression is performed through minimisation of eq 5.5, solving for the projection matrix $\mathbf{R}^k$ and bias term $\mathbf{b}_k$ in the following:

$$\operatorname*{arg\,min}_{\mathbf{R}^k, \mathbf{b}_k} \sum_{\mathbf{I}^i} \sum_{\mathcal{H}_i^k} \left\| \Delta\mathcal{H}_i^k - \mathbf{W}_t \phi_i^k - \mathbf{b}_k \right\|^2 \tag{5.5}$$

Cascaded regression was previously used in facial landmark estimation by [130] where sampling during training used Gaussian noise added to landmark locations. However, this assumes a normal distribution. Instead, direct sampling of the CDO is performed. As RBS is deterministic, a random perturbation is added to the initial tracking state, and the CDO can then be used to generate many training samples.

For such a nonlinear problem, cascaded regression accuracy can be improved by limiting the offset distance used in testing. Rather than applying each regression tier for the full residual error $\Delta\mathcal{H}$, a fractional update is used. Each regressor evaluates half $\Delta\mathcal{H}$. This prevents the first regressor attempting to model the complete error. This increases the number of steps required but reduces instability.

5.5 Parameter Selection

The following section discusses the parameters used in pose estimation and their optimisation over an unseen user from the ICVL dataset. One of the users originally in the training set was removed to serve as a validation set, providing 13,385 images. There are several parameters that can be optimised for the RDF. The change in RDF performance was evaluated using the classification accuracy in the previous chapter. Those parameters are summarised in the following. During training 2000 different splitting criteria were evaluated at each node and a random sample of 1000 pixels was used from each image. The ideal forest depth was $d = 20$. Deeper forests increase evaluation time, with a limited gain in performance. The addition of model fitting and the constraint of real-time processing limited the forests classification of 1000 points to under 2ms. The optimal maximal radius of θ was found to be 60, allowing the features to sample across the hands width.

5.5.1 Maximal Feature Radius

The maximum radius of the features used in RER can also be adjusted. The results for adjusting the radius for each tier k of regression can be seen in Figure 5.8. The optimal feature radius is 20 for each tier and is more stable at each subsequent tier. The number of tiers also impacts performance and their convergence rate can be seen to lessen after the second tier, hence we only use two tiers at runtime.

The RER is intended to recover inaccuracies in the model fitting rather than estimate the entirety of the hand. This limited radius also has an impact on the number of features that are sampled. With a smaller radius, fewer features should be required to maintain the sampling density.

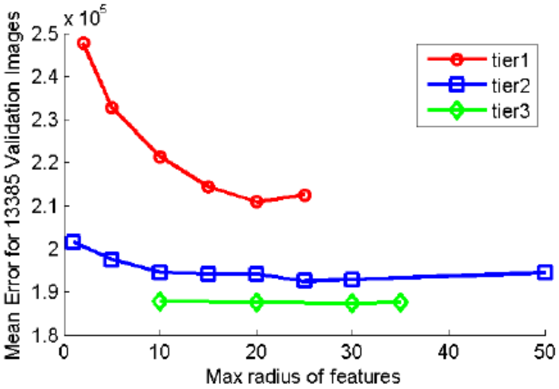


Figure 5.8: Parameter tuning of maximum feature radius when computing pose offset using cascaded linear regression. The reduction in error based on the number of tiers can also be seen, showing less improvement with each additional regressor.

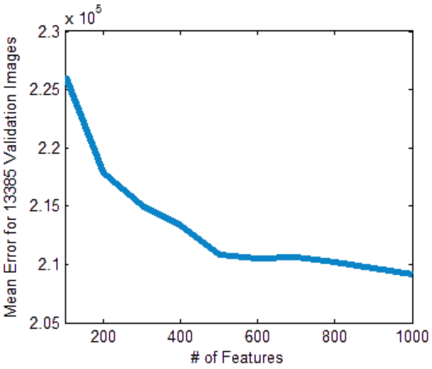


Figure 5.9: The impact on performance when changing the number of features. The performance begins to plateau when the number of features increases above 500.

5.5.2 Number of Features

The cascaded regression model uses a number of features captured at each part. The graph shown in Figure 5.9 demonstrates the change in performance versus the number of features captured. It can be seen that performance plateaus at 500 features. The reduced number of features over those needed in the forest is attributed to the proximity of the features.

5.6 Experiments

The following section details the experimental evaluation of pose estimation. The evaluation includes a qualitative and quantitative part comparing our approach to existing state of the art approaches. A comparison of runtime performance is also conducted, highlighting the highly efficient performance using only a single CPU thread.

The data used in the following experiments are from the ICVL and NYU datasets allowing direct comparison against the state of the art methods of Latent Regression Forests [114] and CNNs [120]. An implementation of Keskin [42] is also evaluated to demonstrate the impact of optimising a generative model over forest segmentation alone. A qualitative evaluation is then provided against Melax [67].

5.6.1 NYU dataset

The evaluation dataset of NYU is labelled using an optimised hand model. There are 36 landmarks, that correspond to a number of locations over their model. The results of NYU's CNN are provided for a subset of these positions. This subset differs from those which correspond to the positions available from the CDO model. This does not allow direct comparison against all of the parts, but rather the overlapping subset. This includes the centre of the hand, three points on the thumb and a point at the end of each finger (indexed as 1, 2, 3, 4, 7, 10, 13, 16). The CNN's result does not produce an estimate for depth, due to their approach regressing to an image location rather than a 3D position. In order to compute error in 3D, the depth for each of their parts is inferred from the depth of the hand. This is also the case for regressed locations that do not reside on the hand (e.g. missing depth). The graph presented in figure 5.10 demonstrates the performance of CDO without part refinement, compared against NYU's CNN [120] and an implementation of mean shift [42].

The performance of the CDO is considerably improved over the mean shift method. This demonstrates the importance of explicitly modelling the Kinematic constraints. The variance between consecutive frames is also reduced, due to the incorporation of temporal modelling. The performance is also comparable with that of NYU's approch [120], which is performing direct regression.

5.6.2 ICVL dataset

Ground truth labels are acquired using automatic means, using the approach of Melax. These landmarks were then manually corrected, with entirely erroneous frames rejected. The landmarks consist of 16 3D part centres and have a depth that is internal to the surface. There are over 20,000 labelled frames captured across 12 users which are synthetically rotated, providing over 300,000 training examples. There are two evaluation sequences for unseen users, each consisting of over 700 frames of labelled poses, again exhibiting challenging interaction poses and transitions.

Concerns regarding the data include temporal sampling and landmark accuracy. Frames were sampled at every third frame, both in the training and test sequences. This increases the difficulty for model-based approaches such as that proposed and Melax's as there is a larger transition between frames. There is also noise present demonstrated in figure 5.11 where it is quantified with a naive measure by calculating the number of landmarks that are positioned outside of hand's contour on a per frame basis. There is also error observed for labelled points inside the contour shown in figure 5.12. Consistent errors in landmark accuracy are also likely to be of benefit to direct regression methods, which can learn the errors present in the training data. During evaluation, this can lead to inaccurate localisation which is consistent with the ground truth.

We evaluate the accuracy of our approach in comparison with accuracy of the state of the art method of Tang which performs the regression of part locations through a hierarchical tree structure. The metrics used are the mean part error for the hand for each frame, and the cumulative mean error as seen in Figure 5.13. The approach exhibits considerably less inter-frame noise, which is again attributed to the use of a model which is temporally coherent, and improves on simply smoothing the resulting detection. This offers a more realistic users experience allowing fine control with less jitter, which is important for a natural interface.

The cumulative mean error shows improvement in accuracy across sequence 1 with a 24% improvement in part localisation which we attribute to the use of cascaded regression. This is confirmed when comparing performance with and without the residual error regression.

The graph in Figure 5.15 demonstrates the impact of residual error regression and shows both the inter-frame and cumulative error over sequence 1. The reduction in cumulative error demonstrates consistent improvement on the unseen data, on average reducing the mean part error by 4.94mm across the sequence which is a 32% decrease in error. Closer inspection of frame wise error (Figure 5.15) shows there are limited instances where regression deteriorates performance. The second sequence shows similar performance to that of Tangs. We attribute this to the presence of faster gestures, which are harder to track due to the temporal sampling.

The following discusses the qualitative comparison with the approach of Melax [67] which is the first approach to utilise RBS. The inclusion of heuristics allows Melax's approach to track a range of poses. Tracking is fast with real-time frame rates and suffers little lag. However in instances of failure, the model becomes trapped in invalid poses, resulting in unrecoverable tracking failure. One such example can be seen in Figure 5.16b while below (fig5.16d) shows our model is less susceptible to local minima. For Melax's approach, recovery of such errors is performed using finger detection, requiring the user to form an initialisation pose of splayed fingers. The CDO approach offers less intrusive reinitialisation as the RDF provides detection continually, allowing seamless recovery from local minima. The integration of prior data in the approach also allows application specific gestures to be trained.

A number of poses that result from the CDO method are illustrated in figure 5.17 and 5.18. The images show our robust localisation against the sequences provided in the ICVL dataset as well a live version of the system, which includes poses of increased difficulty. Several failure cases attributed to local minimum can be seen for the approach of Melax, while CDO performs with good estimation. An accompanying video that demonstrates the temporal performance can be accessed at `https://www.youtube.com/watch?v=Cz4jQi13e0s` which includes comparison with Tang and Melax.

The combination of the approaches allows rapid optimisation of the hand model. Table

5.1 compares the run time performance using a single CPU thread against existing approaches.

Method	**Oikonomidis** [72]	**Sharp** [95]	**Schmidt** [93]	**Qian** [85]	
Device(# threads)	GPU	GPU	GPU	CPU(4)	
Frame Rate(fps)	20	30	30	25	
Method	**Keskin** [43]	**Xu** [126]	**Melax** [67]	**Tang** [114]	**Proposed**
Device(# threads)	CPU(1)	CPU(1)	**CPU(1)**	**CPU(1)**	**CPU(1)**
Frame Rate(fps)	8	12	**60**	**63**	**40**

Table 5.1: Table demonstrating the real time performance of contending approaches. The device used for computation and the frame rate of each method is quoted. Those highlighted demonstrate real time performance using a single threaded implementation.

Figure 5.19 shows the steps taken during error regression in isolation, highlighting its ability to converge on the true part location following CDO. The refined locations provide improved accuracy against model discrepancies. The final pose estimation system is shown for several example poses in Figure 5.20.

5.7 Conclusions

This work presented an approach for hand pose estimation that utilised a combination of techniques, seeking to reduce their mutual failure cases. Through training of a RDF, a region based correspondence can be learned from previously observed hand data. This segmentation provides the assignment for point to surface constraints, allowing a realistic hand model to be fitted to the observation data. Operating as a ICP based method, the minimisation is fast to converge with the observed point cloud. The model provides structural information of the hand, enforcing kinematic limitations and hierarchical constraints, ensuring only natural poses are evaluated. Self-intersection is prevented through the application of collision based constraints, which served to drive intersecting bodies apart. The model fitting is conducted using a Rigid Body Simulation in a Newtonian formalisation, realistically modelling changing poses. Minimisation is initialised using the previous frame's estimate and motion state to incorporate temporal information. In previous approaches, such initialisation would lead to model fitting becoming trapped in local minima. To recover from such failures, approaches depended on manual or fingertip reinitialisation. However, the use of segmentation provides a continuous detection at each frame, allowing the tracking to successfully recover the hand from a range of challenging poses. This allows graceful recovery of tracking, which is important for natural interaction.

In fitting a generic hand model, accuracy is typically limited, due to variation in the shape and proportions of the hand. For this reason, we proposed the use of linear regression that samples from the residual error between the proposed model configuration and the hand's appearance. A correction vector, projected from a high dimensional feature space is then applied iteratively, refining the optimisation's resulting pose, with a mean error of approximately 10mm across the hand's parts.

The presented system provides state of the art performance over three challenging sequences, demonstrating the ability to generalise to new users. Providing real-time tracking with limited computing resources demonstrates potential use in embedded applications and general computing.

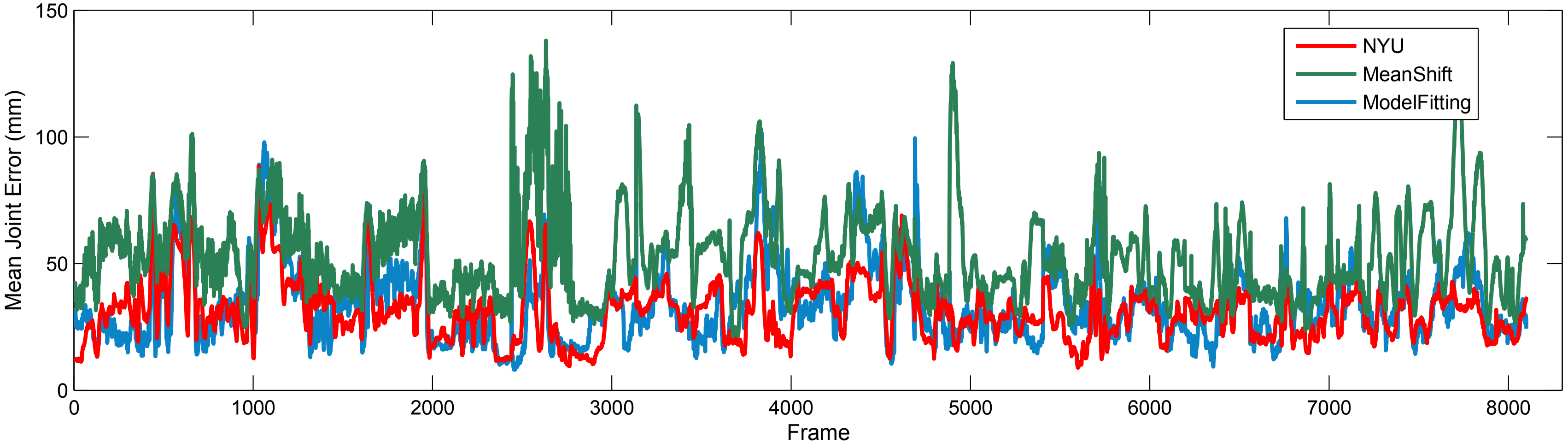


Figure 5.10: The part error over NYU's test set, demonstrating similar performance to the CNN approach. The graph also shows the result of a mean shift method, which shows a large error demonstrating the improvement in using CDO.

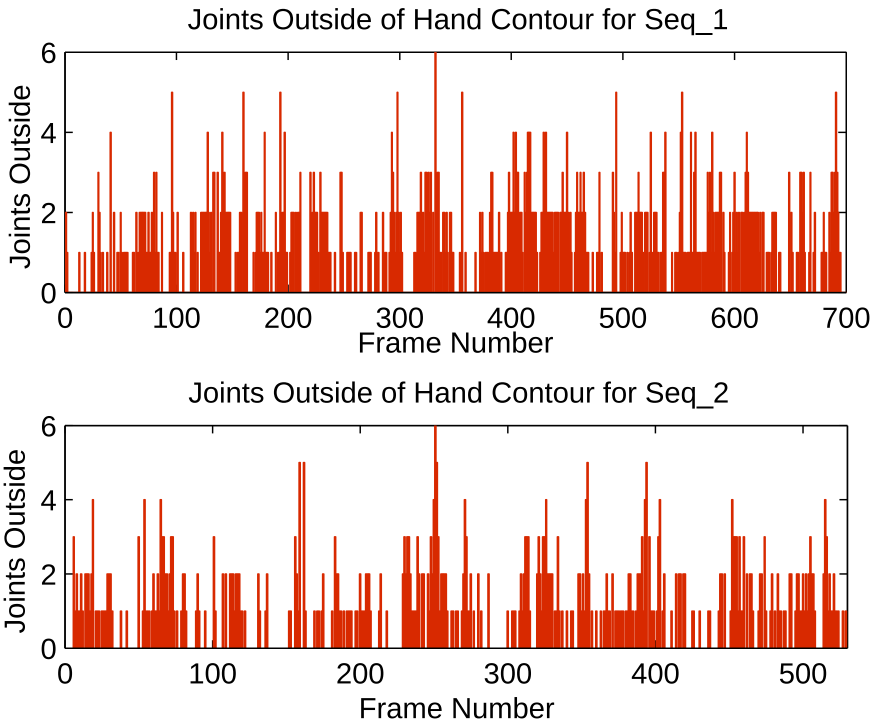


Figure 5.11: Graph showing labelling error in ground truth (GT) of test data. Measured using parts outside of the hand contour in sequence 1 and sequence 2. This error is due to the test sequence having been labelled using automatic means.

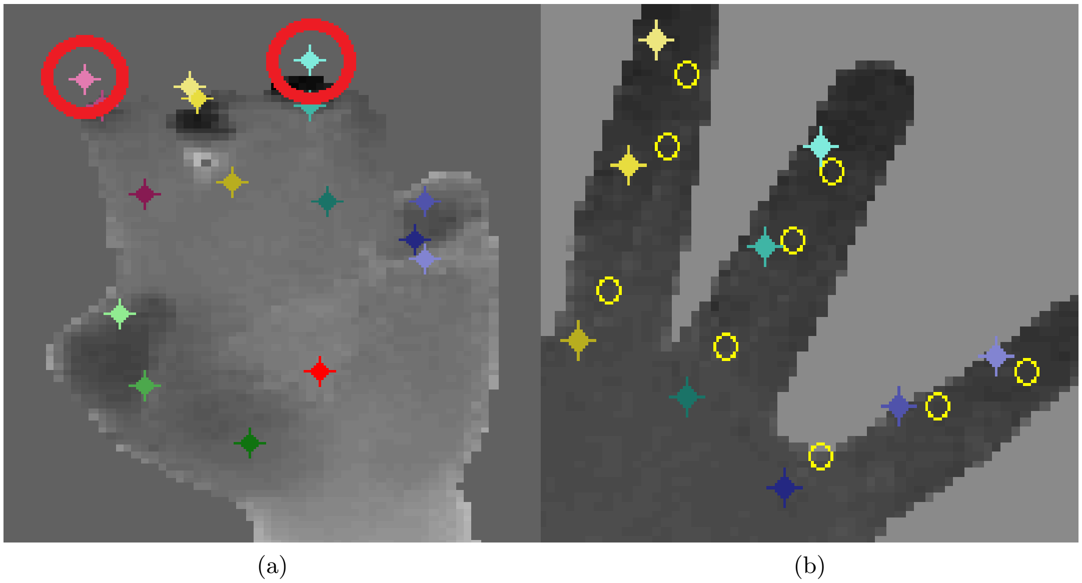


Figure 5.12: The coloured markers show the ground truth provided by Tang, labelled autonomously. (a) Invalid GT points that reside outside of the hand's contour are highlighted. (b) Inaccurate GT points at are inside of the hand's contour. Yellow circles represent the result from the CDO approach, showing better then GT performance.

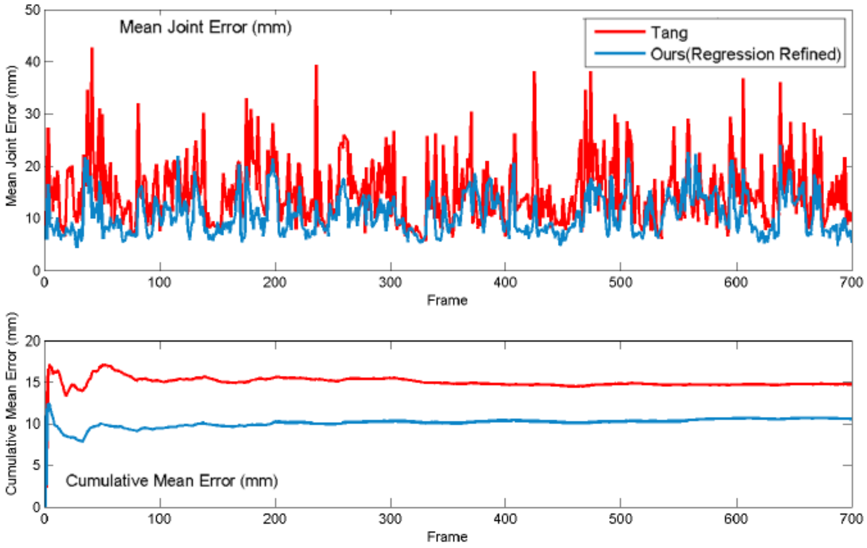


Figure 5.13: Evaluation over sequence 1 from ICVL comparing the per frame mean part error, and it's cumulative moving average.

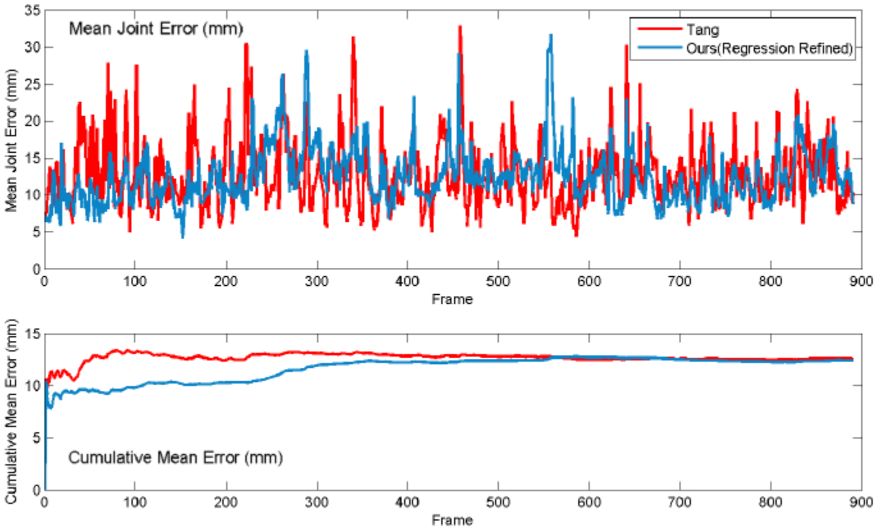


Figure 5.14: Evaluation over sequence 2 from ICVL comparing the per frame mean part error, and it's cumulative moving average.

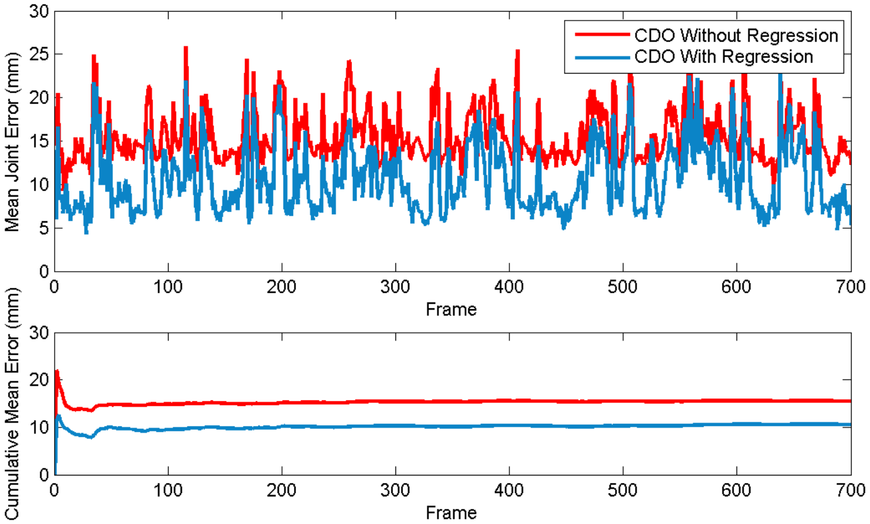


Figure 5.15: Error of hand pose estimation with and without the use of cascaded linear regression.

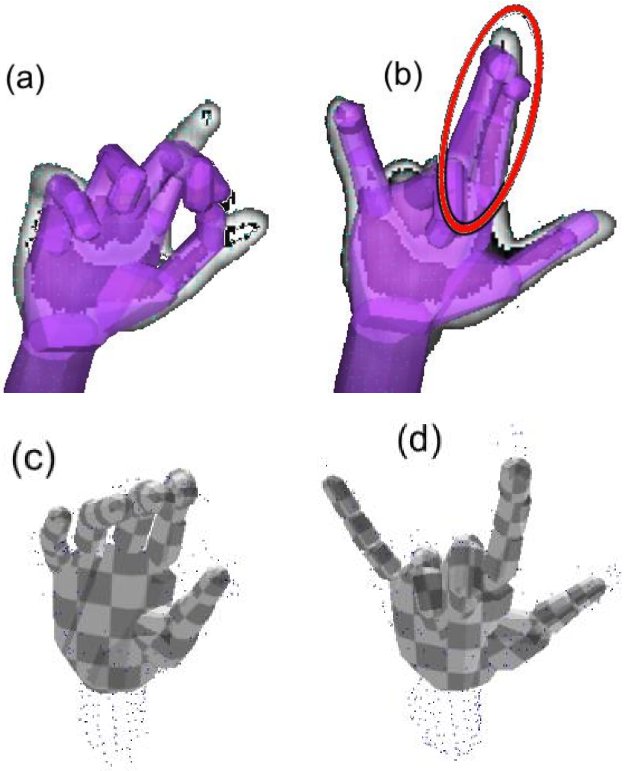


Figure 5.16: Example showing failure instances of Melax's approach. Comparing Melax's method [67](a,b) and CDO(c,d), the model in (a) fails to fully converge with the depth. The middle finger is incorrectly attached to the index finger forming a local minima (b). The proposed model optimisation over a similar sequence (c,d) does not suffer from local minima.

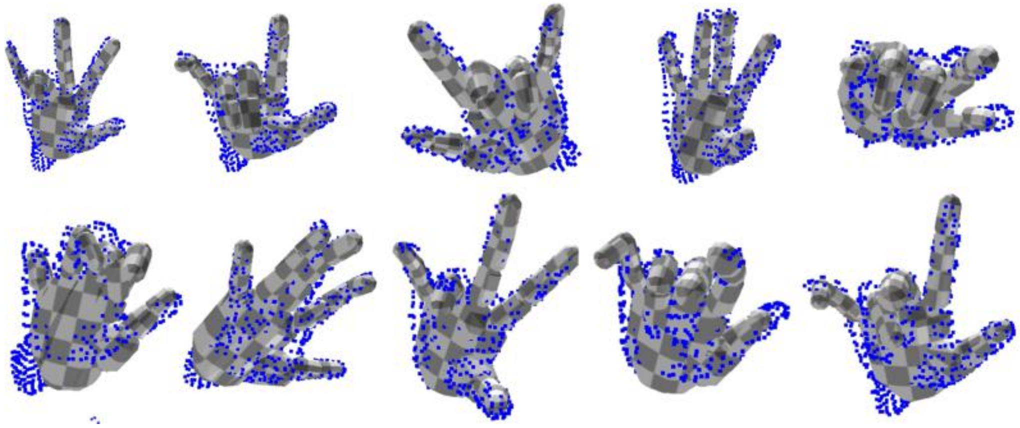


Figure 5.17: Qualitative Evaluation: examples of the combined descent method fitting to point cloud data. The kinematic model prevents self-intersection between fingers and provides realistic results.

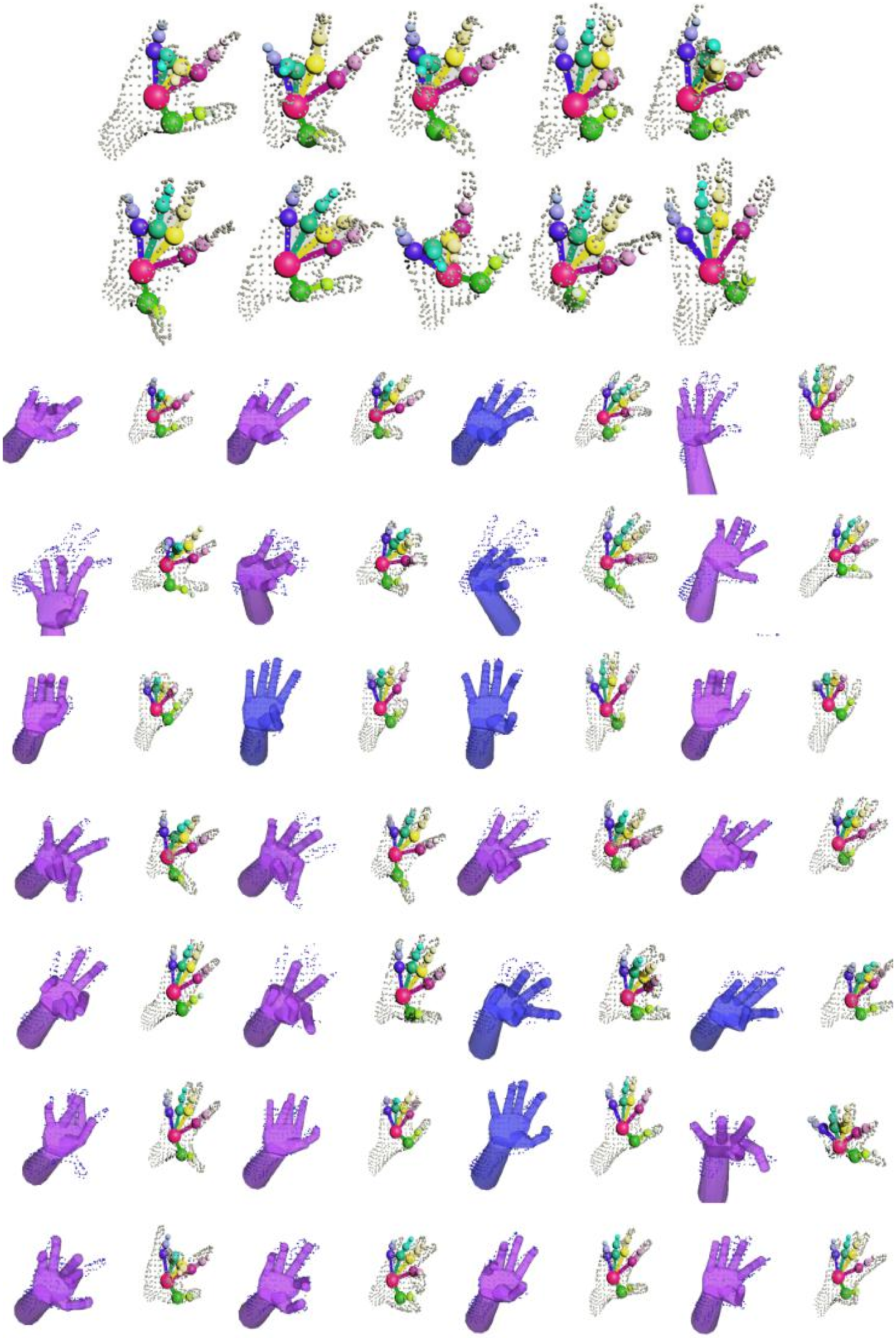


Figure 5.18: The result of model fitting and part refinement for several pose examples, with comparison between Melax (Blue/Purple model) and the refined model over Sequence 1 from ICVL dataset.

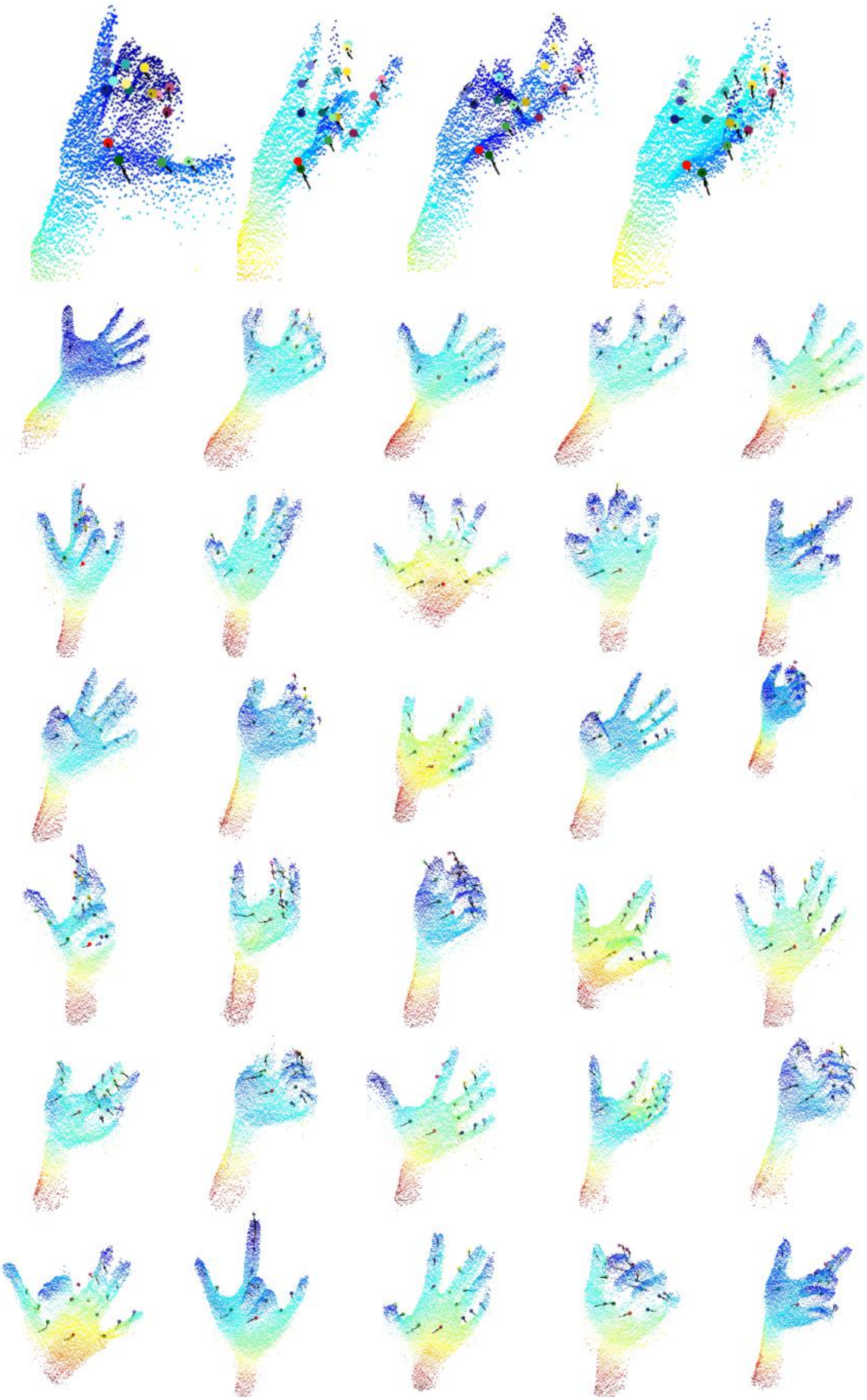


Figure 5.19: Linear regression provides an update vector represented by the arrows at each part. The final proposed part locations are labelled as coloured points.

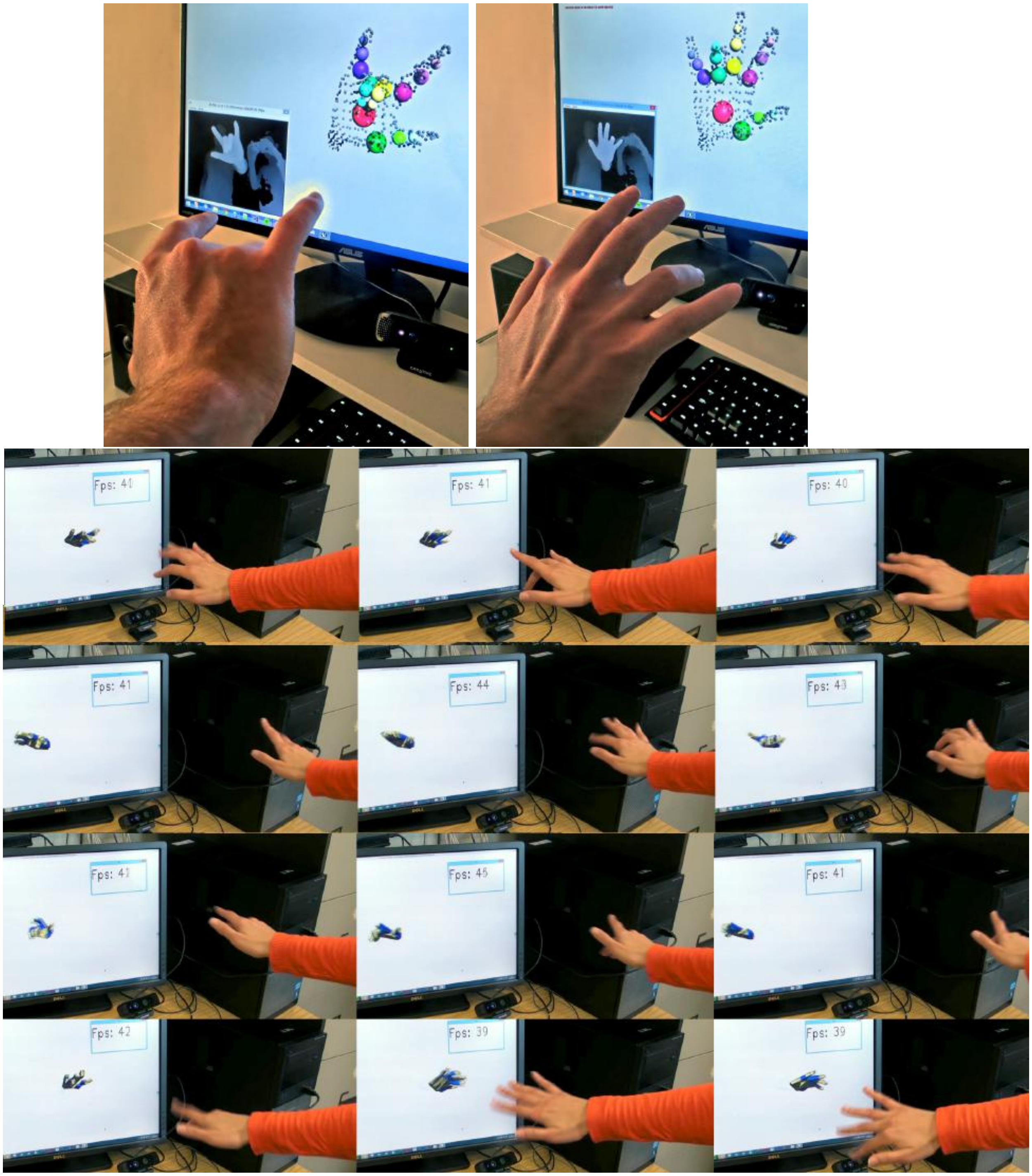


Figure 5.20: Demonstration of the hand pose estimation method in a live capture.

Chapter 6

Discussion

This thesis explored techniques for the challenging problem of hand pose estimation using depth, motivated by recent developments in body pose estimation.

The first chapter of this thesis sought to resolve the fingertips of the hand in 3D space. The objective was to allow multi-touch interaction, without the need for contact with a physical surface. Such technology has the potential to facilitate computing in new exciting applications.

This method used geodesic distance to localise extrema on the surface of the hand. The search was performed using a graph-based technique that searched for multiple extrema. These were then filtered and tracked using a bank of Kalman filters in three-dimensional space. A high degree of accuracy was achieved, with approximately 5mm error per fingertip. This formed the Multi-touchless approach which was combined with simple gesture recognition allowing a real-time contact-less interface for large-scale visualisation of multimedia data.

Identifying which finger is presented was not addressed. However, being able to distinguish between different fingers has the potential to expand the range of possible applications. However, this is challenging due to the similar appearance of fingers and can only be addressed with a structural understanding of the hand, which motivated the remainder of the thesis.

The second technical chapter investigated the use of a Randomised Decision Forest

(RDF). A discriminative approach which can provide the segmentation of the hand, given only a single depth image. This region-based segmentation then provides the structural information required for resolving the hands pose. Most importantly the prediction could be computed in real-time.

The forest itself was trained to model varying hand shapes using an extensive datasets that captures the hands complex structure. Using simple low-level features, each decision tree in the forest is able to implicitly model spatial context. The use of multiple trees allowed the forest to generalise to unseen users, which is required for robust performance.

Acquiring the large amounts of training data needed to generalise is a challenging process, due to the extensive range of possible hands poses. Prior approaches typically rely on the use of synthetic data to provide the majority of data but such data is difficult to generate with the realism required to accurately model real data. This thesis presented an automated means of labelling large quantities of data using a coloured glove and a colour-based GMM classifier. To circumvent misclassification due to challenging lighting, a relative probability measure was computed. This allowed forest training to sample those pixels which are classified with a high confidence.

These techniques were then applied to the challenge of recognising ASL fingerspelling. The corresponding letter for each hand shape was stored in the existing tree structure, removing the need for additional learning, required by previous approaches. This reduced the complexity during evaluation, providing improved performance at test time. The forests classification accuracy was plotted over the course of several sequences which demonstrated a large variance between successive frames. This is attributed to the lack of temporal information, which the following chapter sought to incorporate, through the application of a generative model.

The final technical chapter investigated the application of a generative model formulated as an articulated body simulation. This allowed the incorporation of the motion state which provided temporal information reducing the noise between successive frames. The model also explicitly enforced kinematic constraints, restricting the prediction to realistic poses, which was seen in the improved performance over mode finding approaches. Additionally the model prevented self-intersection improving the boundary conditions between neighbouring fingers and again promoted realism.

This combined optimisation method sought to reduce the mutual failure cases between either approaches alone and provides a high degree of accuracy in localising the joints of the hand. However, it was demonstrated that using a generic model that did not adapt to the user was a limitation of the approach. Optimising the proportions of the model is not tractable, and instead a discriminative technique was employed to refine the joint positions. Each joint was updated through linear regression. This corrects the joint locations and was seen to dramatically improve performance on the test sequences provided by Tang.

There is great potential for the future development of hand pose estimation. The field has progressed quickly since the advent of depth but is still in its infancy.

There is huge potential for hand pose estimation in the field of Sign Language Recognition (SLR) beyond fingerspelling. However, there are increased challenges in estimation due to the rapid motion of signs and the range of hand shapes presented.

Employing other learning techniques could also prove beneficial, providing better partitioning of the hand. One such method could be training a discriminative model to regresses a pixel's position on the surface of the hand. This could be used to provide a dense surface correspondence that does not suffer from the boundary errors seen in chapter 4.

It is possible that temporal information could be modelled in the discriminative approach. This could be achieved by sampling features from the previous frames. The motion between successive frames will increase the complexity of the model, which would require an technique capable of learning large complex relationships such as CNNs.

During model fitting the proportions of the model could be optimised over the course of the sequence. This would reduce the need for the RER and improve the fitting process. There will also be an increase in the complexity of the optimisation, however updates in the models proportions could be regressed by the forest stage.

There are many possible solutions to the challenge of hand pose estimation. With improved accuracy and runtime performance, hand interaction could be applied to many applications where gesture control offers improve user experience such as games, film and ultimately virtual reality.

Bibliography

[1] Irene Albrecht, Jörg Haber, and Hans-peter Seidel. Construction and Animation of Anatomically Based Human Hand Models. *SIGGRAPH*, 2003.

[2] Yali Amit, Donald Geman, and Kenneth Wilder. Joint induction of shape features and tree classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1300–1305, 1997.

[3] Antonis Argyros and Manolis Lourakis. Vision-based interpretation of hand gestures for remote control of a computer mouse. *European Conference on Computer Vision (ECCV)*, 2006.

[4] Andreas Aristidou and Joan Lasenby. Motion Capture with Constrained Inverse Kinematics for Real-Time Hand Tracking. *Communications, Control and Signal Processing*, (March):3–5, 2010.

[5] Vassilis Athitsos and Stan Sclaroff. Estimating 3D hand pose from a cluttered image. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2, 2003.

[6] Andreas Baak, Muller Meinard, Gaurav Bharaj, Hans-peter Seidel, and Christian Theobalt. A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera. *Consumer Depth Cameras for Computer Vision*, Internatio, 2011.

[7] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion Capture of Hands in Action Using Discriminative Salient Points. In *European Conference on Computer Vision (ECCV)*, 2012.

[8] H. Benko, E.W. Ishak, and S. Feiner. Cross-dimensional gestural interaction techniques for hybrid immersive environments. *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, 2005:209–217, 2005.

[9] Richard Bowden, David Windridge, and T Kadir. A linguistic feature vector for the visual interpretation of sign language. In *European Conference on Computer Vision (ECCV)*, 2004.

[10] Matthieu Bray, Esther Koller-meier, Muller Pascal, Luc Van Gool, and Nicol Schraudolph. 3D Hand Tracking By Rapid Stochastic Gradient Descent. *European Conference on Visual Media Production (CVMP)*, 2004.

[11] L Breiman. Random forests. *Machine learning*, pages 5–32, 2001.

[12] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, 2002.

[13] Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. FingARtips Gesture Based Direct Manipulation in Augmented Reality. *Computer Graphics and Interactive Techniques (ACM)*, 1(212):212–221, 2004.

[14] Bulletphysics.org. Real-Time Physics Simulation, Accessed: 2014.

[15] Alexander Buryanov and Viktor Kotiuk. Proportions of Hand Segments. *International Journal of Morphology*, 28(3):755–758, 2010.

[16] Chia-Ping Chen, Yu-Ting Chen, Ping-Han Lee, Yu-Pao Tsai, and Shawmin Lei. Real-time hand tracking on depth images. *Visual Communications and Image Processing (VCIP)*, (1), 2011.

[17] Chin-Seng Chua, Haiying Guan, and Yeong-Khing Ho. Model-based 3D hand posture estimation from a single 2D image. *Image and Vision Computing*, 20(3):191–202, mar 2002.

[18] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision Forests for Classification , Regression , Density Estimation , Manifold Learning and Semi-Supervised Learning. *Microsoft Research Cambridge, Tech. Rep. MSRTR*, 2011.

[19] cyberglovesystems.com CyberGloveSystems LLC, Accessed: 2015.

[20] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-Based 3D Hand Pose Estimation from Monocular Video. *IEEE transactions on pattern analysis and machine intelligence*, pages 1–15, feb 2011.

[21] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1999.

[22] Quentin Delamarre and Olivier Faugeras. Finding pose of hand in video images: A stereo-based approach. *Proceedings - 3rd IEEE International Conference on Automatic Face and Gesture Recognition, FG 1998*, pages 585–590, 1998.

[23] Laura Dipietro, Angeloi Sabatini, and Paolo Dario. A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 2008.

[24] Paul Doliotis, Vassilis Athitsos, Dimitrios Kosmopoulos, and Stavros Perantonis. Hand shape and 3D pose estimation using depth data from a single cluttered frame. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7431 LNCS(PART 1):148–158, 2012.

[25] Sylvia Dominguez, Trish Keaton, and Ali H Sayed. A Robust Finger Tracking Method for Multimodal Wearable Computer Interfacing. *Transactions on Multimedia*, 8(5):956–972, 2006.

[26] Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52–73, oct 2007.

[27] Sean Ryan Fanello, Tim Paek, Cem Keskin, Shahram Izadi, Pushmeet Kohli, David Kim, David Sweeney, Antonio Criminisi, Jamie Shotton, and Sing Bing Kang. Learning to be a depth camera for close-range human capture and interaction. *ACM Transactions on Graphics*, 33(4):1–11, 2014.

[28] Litong Feng, Lai-Man Po, Xuyuan Xu, Ka-Ho Ng, Chum-Ho Cheung, and Kwok-Wai Cheung. An adaptive background biased depth map hole-filling method for Kinect. *Industrial Electronics Society, (IECON)*, pages 2366–2371, 2013.

[29] Ziyong Feng, Shaojie Xu, Xin Zhang, Lianwen Jin, Zhichao Ye, and Weixin Yang. Real-time fingertip tracking and detection using Kinect depth sensor for a new writing-in-the air system. *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service - ICIMCS '12*, page 70, 2012.

[30] Valentino Frati and Domenico Prattichizzo. Using Kinect for hand tracking and rendering in wearable haptics. *World Haptics Conference (WHC)*, 2011.

[31] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2188–2202, 2011.

[32] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Thrun Sebastian. Real-time human pose tracking from range data. *European Conference on Computer Vision (ECCV)*, pages 1–14, 2012.

[33] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, apr 1988.

[34] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–422, 2011.

[35] Haiying Guan, Jae Sik Chang, Feris Rogerio S, and M. Turk. Multi-view Appearance-based 3D Hand Pose Estimation. *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 154–154, 2006.

[36] Georg Hackenberg, Rod McCall, and Wolfgang Broll. Lightweight Palm and Finger Tracking for Real-Time 3D Gesture Control. *Virtual Reality Conference (VR)*, 2011.

[37] Henning Hamer, Konrad Schindler, Esther Koller-Meier, and Luc Van Gool. Tracking a hand manipulating an object. *Computer Vision*, 2009.

[38] Chris Harrison, Hrvoje Benko, and Andrew Wilson. OmniTouch : Wearable Multitouch Interaction Everywhere. *User interface software and technology (ACM)*, pages 441–450, 2011.

[39] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[40] Wolfgang Hürst and Casper Van Wezel. Gesture-based interaction via finger tracking for mobile augmented reality. *Multimedia Tools and Applications*, 62(1):233–258, 2013.

[41] Cem Keskin, Ayse Erkan, and Lale Akarun. Real Time Hand Tracking and 3D Gesture Recognition for Interactive Interfaces Using Hmm. In *ICANN/ICONIPP*, 2003.

[42] Cem Keskin, Furkan Kirac, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1228–1234. IEEE, nov 2011.

[43] Cem Keskin, F Kraç, YE Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. *Computer VisionECCV 2012*, pages 852–863, 2012.

[44] Kim, O Hilliges, S Izadi, Butler, J Chen, Iason Oikonomidis, and P Olivier. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. *Proceedings of UIST 2012*, pages 167–176, 2012.

[45] Furkan Kirac, Yunus Emre Kara, and Lale Akarun. Hierarchically constrained 3D hand pose estimation using regression forests from single frame depth data. *Pattern Recognition Letters*, sep 2013.

[46] Peter Kontschieder, Pushmeet Kohli, Jamie Shotton, and Antonio Criminisi.

GeoF: Geodesic Forests for Learning Coupled Predictors. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 65–72, jun 2013.

[47] Philip Krejov and Richard Bowden. Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7. IEEE, apr 2013.

[48] Philip Krejov, Andrew Gilbert, and Richard Bowden. A Multitouchless Interface. *Computer Graphics and Applications, IEEE*, 2014.

[49] Ana Kuzmanic and Vlasta Zanchi. Hand shape classification using DTW and LCSS as similarity measures for vision-based gesture recognition system. In *EUROCON 2007 - The International Conference on "Computer as a Tool"*, pages 264–269. IEEE, 2007.

[50] Nikolaos Kyriazis and Antonis Argyros. Scalable 3D Tracking of Multiple Interacting Objects. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE*, 2014.

[51] Hervée Lahamy and Derek Litchi. Real-time hand gesture recognition using range cameras. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences [on CD-ROM]*, page 38, 2010.

[52] Leap Motion Inc. leapmotion.com, Accessed: 2012.

[53] Sung Uk Lee and Isaac Cohen. 3D hand reconstruction from a monocular view. *Proceedings - International Conference on Pattern Recognition*, 3:310–313, 2004.

[54] Taehee Lee and Hollerer Tobias. Handy AR : Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. *Wearable Computers*, 2007.

[55] Yi Li. Hand Gesture Recognition Using Kinect. *Software Engineering and Service Science (ICSESS)*, pages 196–199, 2010.

[56] Hui Liang, Junsong Yuan, Senior Member, and Daniel Thalmann. Resolving Ambiguous Hand Pose Predictions by Exploiting Part Correlations. *Circuits and Systems for Video Technology*, pages 1–14, 2015.

[57] Hui Liang, Junsong Yuan, and Daniel Thalman. Egocentric hand pose estimation and distance recovery in a single RGB image. *Multimedia and Expo (ICME)*, 2015.

[58] Hui Liang, Junsong Yuan, and Daniel Thalmann. Parsing the hand in depth images. *IEEE Transactions on Multimedia*, 16(5):1241–1253, 2014.

[59] Hui Liang, Junsong Yuan, Daniel Thalmann, and Zhengyou Zhang. Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization. *The Visual Computer*, 29(6-8):837–848, may 2013.

[60] Lifeprint.com. ASL Fingerspelling Image, Accessed: 2015.

[61] John Lin, Ying Wu, and Thomas S Huang. Modeling the Constraints of Human Hand Motion. *Constraints*, pages 121–126 167, 2000.

[62] Alexandros Makris. Model-based 3D Hand Tracking with on-line Hand Shape Adaptation. *British Machine Vision Conference (BMVC)*, pages 1–12, 2015.

[63] Alexandros Makris, Nikolaos Kyriazis, and Antonis Argyros. Hierarchical Particle Filtering for 3D Hand Tracking. *Computer Vision and Pattern Recognition (CVPR)*, pages 8–17, 2015.

[64] Sotiris Malassiotis, Niki Aifanti, and Michael Strintzis. A Gesture Recognition System Using 3D Data. *3D Data Processing Visualization and Transmission*, 2002.

[65] By Shahzad Malik. Real-time hand tracking and finger tracking for interaction. *Department of Computer Science, University of Toronto, Tech. Rep*, 2003.

[66] Polrola Mateusz and Adam Wojciechowski. Real-Time Hand Pose Estimation Using Classifiers. *Computer Vision and Graphics*, 2012.

[67] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3D skeletal hand tracking. In *Graphics Interface (ACM)*, New York, New York, USA, 2013. ACM Press.

[68] David Minnen and Zahoor Zafrulla. Towards Robust Cross-User Hand Tracking and Shape Recognition. *Computer Vision Workshops (ICCV)*, 2011.

[69] Hasan Mokhtar and Mishra Pramod. Real Time Fingers and Palm Locating using Dynamic Circle Templates. *International Journal of Computer Applications*, 41(6):33–43, 2012.

[70] Ulrich. Neumann and Zhenyao Mo. Real-time Hand Pose Recognition Using Low-Resolution Depth Images. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006.

[71] Natalia. Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Hand segmentation with structured convolutional learning. *Asian Conference on Computer Vision (ACCV)*, 2014.

[72] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient Model-based 3D Tracking of Hand Articulations using Kinect. *British Machine Vision Conference (BMVC)*, 2011.

[73] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis a. Argyros. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. *International Conference on Computer Vision (ICCV)*, 2011.

[74] Iason Oikonomidis, Manolis Lourakis, and Antonis Argyros. Evolutionary Quasi-random Search for Hand Articulations Tracking. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[75] Kenji Oka, Yoichi Sato, and Hideki Koike. Real-Time Fingertip Tracking and Gesture. *Computer Graphics and Applications*, 2002.

[76] Zhigeng Pan, Yang Li, Mingmin Zhang, Chao Sun, Kangde Guo, Xing Tang, and Steven Zhou. A real-time multi-cue hand tracking algorithm based on computer vision. In *Virtual Reality Conference (VR)*, 2010.

[77] Jun Park and Yeo Lip Yoon. LED-glove based interactions in multi-modal displays for teleconferencing. *Proceedings - 16th International Conference on Artificial Reality and Telexistence - Workshops, ICAT 2006*, pages 395–399, 2006.

[78] Fabrizio Pedersoli, Sergio Benini, Nicola Adami, and Riccardo Leonardi. XKin: an

open source framework for hand pose and gesture recognition using kinect. *Visual Computer*, pages 1–16, 2014.

[79] Son Phung, Abdesselam Bouzerdoum, and Douglas Chai. Skin segmentation using color pixel classification: analysis and comparison. *Pattern Analysis and Machine Intelligence (PAMI)*, 2005.

[80] W. Piekarski and B.H. Thomas. Using ARToolKit for 3D hand position tracking in mobile outdoor environments. *The First IEEE International Workshop Agumented Reality Toolkit,*, (X):2–3, 2002.

[81] Christian Plagemann, Varun Ganapathi, Daphne Koller, and Thrun Sebastian. Real-time Identification and Localization of Body Parts from Depth Images. *Robotics and Automation (ICRA)*, 2010.

[82] Georg Poier, Konstantinos Roditakis, Samuel Schulter, Damien Michel, Horst Bischof, and Antonis A. Argyros. Hybrid One Shot 3D Hand Pose Estimation By Exploiting Uncertainties. In *British Machine Vision Conference (BMVC)*, 2015.

[83] Rudra Poudel, Jose Fonseca, Jian Zhang, and Hammadi Nait-Charif. A Unified Framework for 3D Hand Tracking. *Advances in Visual Computing*, 2013.

[84] Nicolas Pugeault and Richard Bowden. Spelling It Out: Real-Time ASL Fingerspelling Recognition. *International Conference on Computer Vision Workshop (ICCV Workshop)*, 2011.

[85] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and Robust Hand Tracking from Depth. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, jun 2014.

[86] Shuxin Qin, Xiaoyang Zhu, Yiping Yang, and Yongshi Jiang. Real-time hand gesture recognition from depth images using convex shape decomposition method. *Journal of Signal Processing Systems*, 74(1):47–58, 2014.

[87] J.M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. *Proceedings of IEEE International Conference on Computer Vision*, pages 612–617, 1995.

[88] JM Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. *European Conference on Computer Vision (ECCV)*, (May):35–46, 1994.

[89] Xiaofeng Ren, Alexander C Berg, and Jitendra Malik. Recovering human body configurations using pairwise constraints between parts. In *International Conference on Computer Vision (ICCV)*, 2005.

[90] Konstantinos Roditakis and Antonis A. Argyros. Quantifying the Effect of a Colored Glove in the 3D Tracking of a Human Hand. In *Computer Vision Systems*, volume 5815, pages 404–414. 2015.

[91] Javier Romero, Hedvig Kjellström, and Danica Kragic. Monocular real-time 3D articulated hand pose estimation. *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*, pages 87–92, 2009.

[92] Javier Romero, Hedvig Kjellstrom, and Danica Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. *Robotics and Automation (ICRA)*, may 2010.

[93] Tanner Schmidt, Richard Newcombe, and Dieter Fox. DART: Dense Articulated Real-Time Tracking. *Robotics: Science and Systems*, (1), 2014.

[94] Loren Arthur Schwarz, Artashes Mkhitaryan, Diana Mateus, and Nassir Navab. Estimating Human 3D Pose from Time-of-Flight Images Based on Geodesic Distances and Optical Flow. *Automatic Face & Gesture Recognition (FG)*, 2011.

[95] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freeman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. In *Human Factors in Computing Systems (CHI)*, Seoul Korea, 2015.

[96] N. Shimada, K. Kimura, and Y. Shirai. Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera. In *Proceedings IEEE*

ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pages 23–30. IEEE Comput. Soc.

[97] Seymour Shlien. Multiple binary decision tree classifiers. *Pattern Recognition*, 23(7):757—-763, 1990.

[98] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. IEEE, jun 2011.

[99] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, and Andrew Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, dec 2013.

[100] Leonid Sigal, Stan Sclaroff, and Vassilis Athitsos. Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):862–877, 2004.

[101] Georgiana Simion, Vasile Gui, and Marius Otesteanu. Finger Detection Based on Hand Contour and Colour Information. *Applied Computational Intelligence and Informatics (SACI)*, 2011.

[102] Myoung-kyu Sohn, Dong-ju Kim, and Hyunduk Kim. Hand Part Classification Using Single Depth Images. *Asian Conference on Computer Vision (ACCV)*, 2014.

[103] Dan Song, Nikolaos Kyriazis, Iason Oikonomidis, Chavdar Papazov, Antonis Argyros, Darius Burschka, and Danica Kragic. Predicting human intention in visual observations of hand/object interactions. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1608–1615, 2013.

[104] Jie Song, Gabor Soros, Fabrizio Pece, Sean Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air Gestures Around Unmodified Mobile Devices. *User interface software and technology (ACM)*, 2014.

[105] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and Robust Hand Tracking Using Detection-Guided Optimization. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[106] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. *2013 IEEE International Conference on Computer Vision*, pages 2456–2463, 2013.

[107] T Starner and A Pentland. Real-time American Sign Language recognition from video using hidden Markov models. In *Proceedings of International Symposium on Computer Vision - ISCV*, pages 265–270. IEEE Comput. Soc. Press, 1995.

[108] Björn Stenger, P.R.S. Mendonca, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–310–II–315. IEEE Comput. Soc, 2001.

[109] Björn Stenger, Arasanathan Thayananthan, Philip H S Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical Bayesian filter. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1372–84, sep 2006.

[110] Jesus Suarez and Robin R. Murphy. Hand gesture recognition with depth images: A review. *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 411–417, sep 2012.

[111] Erik Sudderth, Michael Mandel, William Freeman, and Alan Willsky. Visual Hand Tracking Using Nonparametric Belief Propagation. *Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, 2004.

[112] Andrea Tagliasacchi, Matthias Schröder, and Anastasia Tkach. Robust Articulated-ICP for Real-Time Hand Tracking. *Computer Graphics Forum*, 34(5), 2015.

[113] Masaki Takahashi. Human Gesture Recognition using 3.5-Dimensional Trajectory Features for Hands-Free User Interface. *Analysis and retrieval of tracked events and motion in imagery streams (ACM)*, 2010.

[114] Danhang Tang, HJ Chang, A Tejani, and TK Kim. Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture. In *CVPR 2014*, 2014.

[115] Danhang Tang, TH Yu, and TK Kim. Real-time Articulated Hand Pose Estimation using Semi-supervised Transductive Regression Forests. In *ICCV*, 2013.

[116] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 103–110, 2012.

[117] Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-Specific Hand Modeling from Monocular Depth Sequences. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 644–651, jun 2014.

[118] Daniel Thalmann, Hui Liang, and Junsong Yuan. 3D fingertip and palm tracking in depth image sequences. *Proceedings of the 20th ACM international conference on Multimedia - MM '12*, page 785, 2012.

[119] Akshaya Thippur, Carl Henrik Ek, and Hedvig Kjellstrom. Inferring hand pose: A comparative study of visual shape features. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013*, 2013.

[120] Jonathan Tompson, Murphy Stein, Yann Lecun, Ken Perlin, and Offline Database. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. In *SIGGRAPH*, 2014.

[121] Dimitrios Tzionas, Abhilash Srikantha, Pablo Aponte, and Juergen Gall. Capturing Hand Motion with an RGB-D Sensor, Fusing a Generative Model with Salient Points. In *Pattern Recognition*, pages 277–289. 2014.

[122] Michael Van Den Bergh and Luc Van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. *Applications of Computer Vision (WACV)*, 2011.

[123] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[124] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *Transactions on Graphics (ACM)*, jul 2009.

[125] Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics*, 32(4):1, 2013.

[126] Chi Xu and Li Cheng. Efficient Hand Pose Estimation from a Single Depth Image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3462, 2013.

[127] Yuan Yao, Yun Fu, and Senior Member. Contour Model-Based Hand-Gesture Recognition Using the Kinect Sensor. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(11):1935–1944, 2014.

[128] H S Yeo, B G Lee, and Hyotaek Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*, pages 1–29, 2013.

[129] Chenglong Yu, Xuan Wang, Hejiao Huang, Jianping Shen, and Kun Wu. Vision-Based Hand Gesture Recognition Using Combinational Features. *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 543–546, oct 2010.

[130] Xiang Yu, Zhe Lin, Jonathan Brandt, and DimitrisN. Metaxas. Consensus of Regression for Occlusion-Robust Facial Feature Localization. *Computer Vision ECCV 2014 SE - 8*, 8692:105–118, 2014.

[131] Yu Yu, Yonghong Song, and Yuanlin Zhang. Real Time Fingertip Detection with Kinect Depth Image Sequences. *2014 22nd International Conference on Pattern Recognition*, pages 550–555, 2014.

[132] Zhou Ren, Jingjing Meng, Junsong Yuan, Zhou Ren, Jingjing Meng, and Junsong Yuan. Depth camera based hand gesture recognition and its applications in Human-

Computer-Interaction. *International Conference on Information, Communications & Signal Processing*, 2011.