Robust Hand Pose Recognition from Stereoscopic Capture

Rilwan Remilekun Basaru

Department of Computer Science

City, University of London

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

City, University of London

June 2018

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 65,000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures. I grant powers of discretion to the City, University of London librarian to allow the dissertation to be copied in whole or in part without further reference to myself (the author). This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgment.

Rilwan Remilekun Basaru
June 2018

Acknowledgements

First, I would like to acknowledge the role of my father, Alhaji Sikiru Babatunde Basaru in encouraging and supporting me, both financially and emotionally in achieving this degree. My gratitude is conveyed to my brilliant team of supervisors Dr. Gregory Slabaugh, Dr. Chris Child and Dr. Eduardo Alonso for guiding me through my research studies. Thank you, Greg, for being an inspiration to me, your unwavering support throughout this period and those weekly one-on-one supervision meetings at the beginning.

I would also like to acknowledge the role of the examining team of Dr. Radu Jianu (Chair), Dr. Constantino Carlos Reyes-Aldasoro (internal examiner) and Dr. Philip Edwards (external examiner) for a very informed and educative 4 hours viva.

Lastly, a special thanks to my fellow research students particularly Dr. Muhammad Asad for his extensive input in my research and company during lunch breaks. A special mention to the Computer vision reading group members: Dr. Muhammad Asad, Anfisa Lazareva, Nathan Olliverre, Dr. S. M. Masudur Rahman Al-Arif, Atif Riaz, Greg Slabaugh, Wei Yong, Constantino Carlos Reyes-Aldasoro for there idea-provoking discussing every fortnight.

Abstract

Hand pose is emerging as an important interface for human-computer interaction. The problem of hand pose estimation from passive stereo inputs has received less attention in the literature compared to active depth sensors. This thesis seeks to address this gap by presenting a data-driven method to estimate a hand pose from a stereoscopic camera input, with experimental results comparable to more expensive active depth sensors. The frameworks presented in this thesis are based on a two camera stereo rig capture as it yields a simpler and cheaper set-up and calibration. Three frameworks are presented, describing the sequential steps taken to solve the problem of depth and pose estimation of hands.

The first is a data-driven method to estimate a high quality depth map of a hand from a stereoscopic camera input by introducing a novel regression framework. The method first computes disparity using a robust stereo matching technique. Then, it applies a machine learning technique based on Random Forest to learn the mapping between the estimated disparity and depth given ground truth data. We introduce Eigen Leaf Node Features (ELNFs) that perform feature selection at the leaf nodes in each tree to identify features that are most discriminative for depth regression. The system provides a robust method for generating a depth image with an inexpensive stereo camera.

The second framework improves on the task of hand depth estimation from stereo capture by introducing a novel superpixel-based regression framework that takes advantage of the smoothness of the depth surface of the hand. To this end, it introduces Conditional Regressive Random Forest (CRRF), a method that combines a Conditional

Random Field (CRF) and a Regressive Random Forest (RRF) to model the mapping from a stereo RGB image pair to a depth image. The RRF provides a unary term that adaptively selects different stereo-matching measures as it implicitly determines matching pixels in a coarse-to-fine manner. While the RRF makes depth prediction for each super-pixel independently, the CRF unifies the prediction of depth by modeling pair-wise interactions between adjacent superpixels.

The final framework introduces a stochastic approach to propose potential depth solutions to the observed stereo capture and evaluate these proposals using two convolutional neural networks (CNNs). The first CNN, configured in a Siamese network architecture, evaluates how consistent the proposed depth solution is to the observed stereo capture. The second CNN estimates a hand pose given the proposed depth. Unlike sequential approaches that reconstruct pose from a known depth, this method jointly optimizes the hand pose and depth estimation through Markov-chain Monte Carlo (MCMC) sampling. This way, pose estimation can correct for errors in depth estimation, and vice versa.

Experimental results using an inexpensive stereo camera show that the proposed system measures pose more accurately than competing methods. More importantly, it presents the possibility of pose recovery from stereo capture that is on par with depth based pose recovery.

Notation

For ease of presentation, vectors and matrices are denoted with a boldface lower-case ($\boldsymbol{x}$) and upper-case ($\boldsymbol{X}$) respectively. Vector/matrix transpose are denoted with an upper script T as in $\{\}^T$ whilst the column-order and row-order concatenation are represented as $[\boldsymbol{X}, \boldsymbol{Y}]$ and $[\boldsymbol{X}; \boldsymbol{Y}]$ respectively. Unless explicitly specified, all vectors are assumed to be column vectors e.g. $\boldsymbol{p} = [p_x, p_y, p_z]^T$. A vector where all its elements are one is denoted with $\boldsymbol{i}$, whilst $\boldsymbol{I}$ denotes the identity matrix.

$\mathbb{I}[]$ denotes an indicator function that returns 1 if the argument is true and returns 0 otherwise. Matrix/vector multiplication is denoted as in $\boldsymbol{XY}$, whilst $\times$ between two scalars is used to indicate dimension as in $a \times b$ describes a a-by-b dimensional space or matrix. The equal notation $=$ is used to assign value to variable whilst the $\sim$ indicates the sampling from a probability distribution.

The probability of a variable x is represented as in $Pr(x)$ with $Pr(x, y)$ and $Pr(x|y)$ denoting joint and conditional probability. An unnormalized probability is presented as $\tilde{P}r(x)$.

Chapter 1

Introduction

The aim of this research is to achieve robust hand pose estimation from stereoscopic images, such as those produced by egocentric video cameras. Hand pose estimation has several practical uses, such as human-computer interaction and virtual reality applications [1–4]. This work naturally rests in the field of computer vision, specifically research in articulated object tracking, which has recently attracted increasing attention in the literature [5–9]. This increase in research interest is largely due to recently available commercial depth cameras [10]. Although there are many ways to image the hand and its articulations, two primary approaches that have been employed include conventional RGB cameras and more recently, depth imaging, popularised by commercially available depth sensors like the Microsoft Kinect and ASUS Xtion Pro as well as hand articulation detection products such as the HoloLens and Leap Motion.

Conventional monocular (single RGB) cameras have inherent limitations to their robustness as input to hand pose estimation tasks. These include: a lack of shape information (used to generate significant features and cues for articulation); poor robustness against background clutter; and variance in the output data as a result of changes in ambient condition (brightness/dimness of scene). These shortcomings have led to the advent of active depth cameras. These cameras actively emit electromagnetic (EM) waves towards the scene, probing how far each point in the scene is away from the imaging device. Whilst depth imaging provides good shape information and hence robustness to clutter and changes in ambient conditions, depth sensors present several limitations, including: poor form factor [1]; large energy consumption; poor near distance coverage; and poor outdoor usage. For instance, the stereo camera used in the thesis, Minoru 3D Webcam, has a power consumption of 1.5 Watts [11] whilst the Xbox One Kinect RGBD sensor (used for data collection) has a power rating of approximately 15 Watts. The latter exceeds the power of a standard USB 3 port.

Motivation: The success of the HoloLens is evidence of the continual urge for technology to become an extension of human capability. Hence devices should conform to means of interaction that are natural to humans such as gestures, speech etc. A typical example of these is egocentric based devices. A device is described to be egocentric if it encourages first-person experience and engages with the human from the user's perspective. These devices are a promising approach to improving human-computer interaction and this is evident in that currently, prominent products of major technology companies such as Snap Inc., Google Inc., Facebook Inc. and Microsoft Corporation are first-person interactive devices, e.g. Spectacles, Google Glass 2.0, Oculus Rift and HoloLens respectively. This motivates the need for devices that require

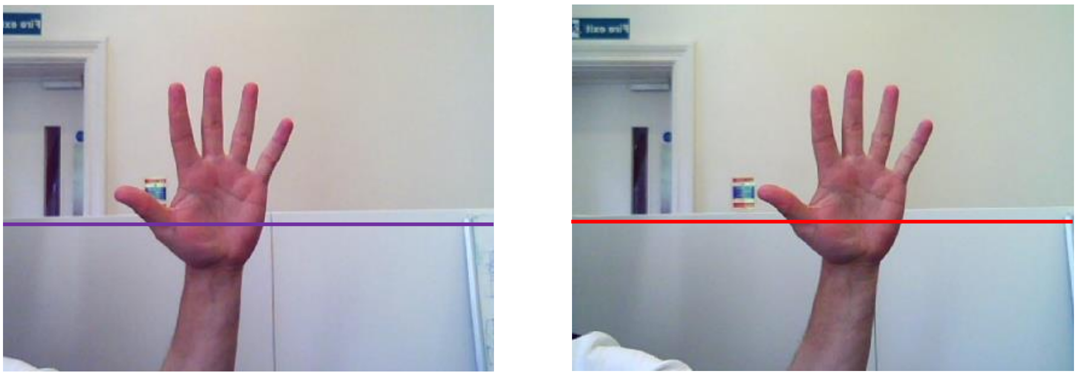


Figure 1.1 The rectified stereo image pair of a hand pose. The purple and red lines indicate respective horizontal line in the left and right stereo captures that share corresponding points.

[1] The form factor of a device explains how large in shape and size the device is. A poor form factor indicates a device too large to the detriment of its portability.

lower energy consumption; better form factor; better indoor coverage; and better near distance coverage.

1.1 Research Goal

The above issues motivate the use of an input system that addresses the disadvantages of the two approaches described above, namely stereo imaging. With stereoscopic input, one is able to achieve close range acquisition, good form factor, and outdoor usage whilst still being able to acquire shape information and maintain robustness to clutter. Hence, this research is based on hand gesture recognition from stereo-optically acquired depth data from egocentric viewpoints, with emphasis on applications such as sign language recognition, virtual/natural interaction, and gaming inputs amongst others. The application of stereo camera to hand pose estimation is based on the concept of *Stereopsis*, which relies on establishing correspondences between two cameras in a stereo capture. This is a challenging task as there is often a lack of consistency

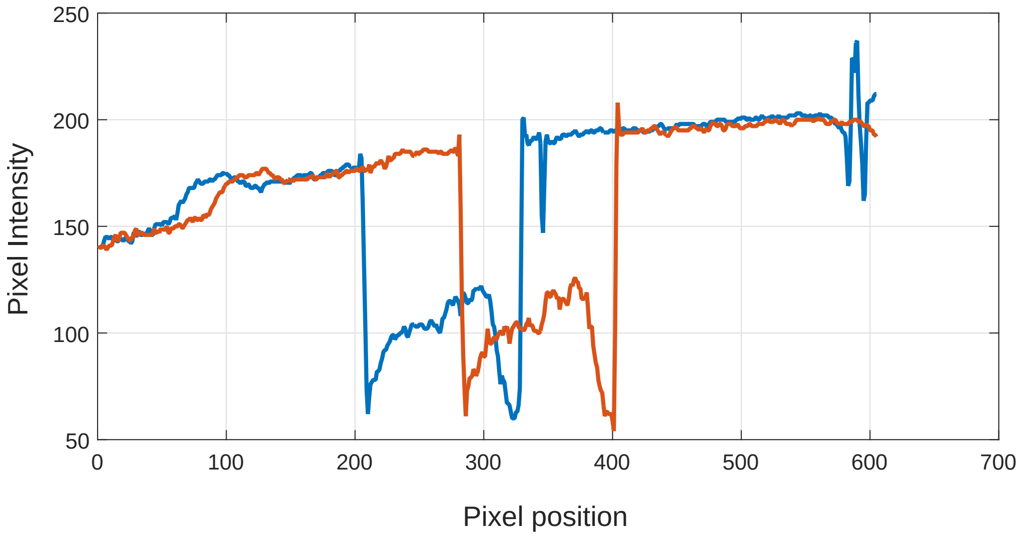


Figure 1.2 Intensity plot of the stereo image pair in Figure 1.1. The red plot indicates the intensity along the red line whilst the blue plot indicates the intensity along the blue line.

between the two stereo cameras as illustrated in Figure 1.1 and 4. Figure 4 shows an intensity plot of the stereo image pair in Figure 1.1. The orange plot indicates the intensity along the red line (in the right stereo image), whilst the blue plot indicates the intensity along the purple line (in the left stereo image). It is apparent that whilst there is consistency in some major variation in intensity, there is a lack of consistency in more subtle intensity variation in both cameras. This makes this research a difficult task, particularly when applied to hand (that tend to have textureless regions). This thesis focusses solely on the problem of single hand depth and pose estimation, which is a real challenge given differences between individuals. Due to time constraints during the PhD and the execution time of the proposed framework, it was not possible to implement the proposed methods in an egocentric view and test outdoors. Also, due to computational complexity, the methods presented in this thesis are not yet capable of real-time performance. Therefore, egocentric and real-time extensions of the work are left for future research. Nonetheless, the solution presented in the succeeding chapters of these thesis, serves as a proof of concept for stereoscopically-based robust hand pose/articulation recovery with performance on-par with RGBD-based systems.

1.1.1 Research Questions

This research aims to answer four questions:

1. How can highly robust depth information be recovered from stereoscopic imaging of hands?

2. How can the problems of texture-less hand regions and radiometric differences be addressed?

3. How can hand pose/articulation be estimated from stereoscopic inputs?

4. How do the results compare to those estimated from depth image inputs?

1.1.2 Research Objectives

Addressing these research questions yields four research objectives, which include:

1. To propose, develop, implement and evaluate hand depth estimation frameworks.

2. To propose, develop and implement a framework for hand pose/articulation estimation from recovered hand depth.

3. To propose a new approach for joint stereo reconstruction and pose estimation of a hand from stereo inputs.

4. To compare the performance of the stereo-based pose estimation approach to RGBD input-based approach

1.2 Research Contributions

The following are the key contributions of this thesis:

1. A new regressive feature selection technique called Eigen Leaf Node Features is presented. In the leaf node of each tree in a Random Forest, this technique factorizes for the posterior probability and regresses the depth using highly discriminant features. Eigen Leaf Node Features is applied to stereoscopic images of hands to learn the mapping between a lower quality disparity estimation and a high-quality groundtruth depth measurement.

2. A machine learning approach to establishing stereo correspondences, by solving a superpixel-based regression problem rather than explicitly minimising a stereo-matching cost function is introduced. Rather than rely on a single cost function or single window size, the proposed method fuses multiple cost functions computed over different window sizes as input to the regressors. Expert trees that learn from different subsets of the data, based on holistic hand features, like skin tone, are proposed.

3. A closed-form non-iterative solution to a Conditional Random Fields-based consolidation of Random Forest Trees predictions is derived.

4. Unlike several approaches to pose estimation from stereo capture that explicitly recover disparity before regressing for the pose in a sequential manner, a joint optimization approach that is robust against potential error in the depth estimation is presented. Thus, this reduces the burden on the pose estimation framework to be robust against erroneous depth recovery. A semi-generative approach that is experimentally proven to work on different sizes and tones of hand without pre-calibration is proposed and implemented.

1.3 Publications

The following peer-reviewed papers have been published:

1. Basaru, R., Child, C., Alonso, E., & Slabaugh, G. (2014). Quantized Census for Stereoscopic Image Matching. In Proc. of the 3DV Conference: Workshop, Dynamic Shape Measurement and Analysis, Dec 2014, Tokyo, Japan.

2. Asad, M., Gentet E., Basaru R., & Slabaugh G. (2015). Generating a 3D Hand Model from Frontal Color and Range Scans. In Proc. of the IEEE International Conference on Image Processing, Sept 2015, Quebec, Canada.

3. Basaru, R., Child, C., Alonso, E., & Slabaugh, G. (2016). HandyDepth: Example-based Stereoscopic Hand Depth Estimation using Eigen Leaf Node Features. In Proc. of the International Conference on Systems, Signals and Image Processing, May 2016, Bratislava, Slovakia.

4. Basaru, R., Child, C., Alonso, E., & Slabaugh, G., (2017). Conditional Regressive Random Forest Stereo-based Hand Depth Recovery. In Proc. of International Conference on Computer Vision: HANDS Workshop, Oct 2017, Venice Italy.

5. Basaru, R., Child, C., Alonso, E., & Slabaugh, G., (2017). Hand Pose Estimation Using Deep Stereovision and Markov-chain Monte Carlo. In Proc. of International Conference on Computer Vision: HANDS Workshop, Oct 2017, Venice Italy.

6. Basaru, R., Child, C., Alonso, E., & Slabaugh, G., (2018). Data-driven Recovery of Hand Depth using Conditional Regressive Random Forest on Stereo Images. IET Computer Vision Journal.

1.3.1 Academic Activities

The student participated in the following activities during the course of the research

1. 3DV 2014 Conference: Workshop, Dynamic Shape Measurement and Analysis (11 December, 2014) - Talk.

2. British Computer Society (BCS) Doctoral Consortium invited talk (22 April, 2015) - Talk.

3. IWISSIP 2016 International Conference (24 May 2016) - Talk.

4. BMVA Technical Meeting: Dynamic Scene Reconstruction 2017 (21 June 2017) - Talk.

1.3.2 Framework and Tools Produced

The following frameworks and tools were produced during the course of the research

1. An OpenGL based tool for hand pose articulation and rendering. This is a C++ implemented rigged hand model that can be used to generate synthetic hand capture of different poses, from different perspectives under different lighting conditions.

2. A MATLAB based testbed for Stereo-matching cost functions against various radiometric difference.

3. C++ code for capturing from RGB-D Sensor, Minoru Stereo camera and Leap-Motion Sensor simultaneously.

4. Calibration and registration framework for simultaneous Minoru and Kinect Sensor capture. This is used to register the depth channel of the RGB-D sensor to the reference stereo camera.

1.4 Thesis Outline

The remaining chapters of this thesis are structured as follows: Chapter 2 presents a brief review of literature that are related to the research topic. Chapter 3 introduces and expatiates on key concepts of stereoscopy, machine learning, and hand pose estimation while Chapter 4 introduces our Eigen-leaf Node Random Forest approach to hand depth recovery from stereo capture. In Chapter 5, we improve upon this by presenting an approach that combines a Conditional Random Field with a Regressive Random Forest. Chapter 6 presents a joint optimization approach to the problem of pose recovery from stereo. We evaluate the three approaches presented in Chapters 4, 5 and 6. A comparison of these approaches is presented in Chapter 7. The work concludes with future work discussed in Chapter 8.

Chapter 2

Literature Review

Over the last decade, hand articulation estimation and tracking, as well as multi-view reconstruction have been popular topics in computer vision and have received considerable attention [12–21]. This chapter gives a review of methods related to those in this research. The literature is categorized into depth recovery from stereo views, general hand articulation recovery, and stereo-based hand pose recovery.

2.1 Stereo Approaches

As discussed in the previous chapter, the recent success in hand pose estimation from active depth sensors has established the significance of shape information as being paramount to hand pose estimation. This, in turn, has motivated the approach of hand depth estimation from stereo capture as disparity recovered from stereo can be used as a precursor for hand pose estimation (see Chapter 3 where the fundamentals of stereo vision are introduced).

Depth estimation from two views has a long and rich history in computer vision and fundamentally relates to establishing correct correspondences between images. There have been several literature surveys pertaining to stereo algorithms [22–25]. Following [23] stereo-matching algorithms are categorized based on four steps:

1. Computing the matching cost.

2. Cost Aggregation.
3. Disparity selection.
4. Disparity refinement. [24]

Each of these four points will discussed in the following subsections.

2.1.1 Matching-cost based Classification

As discussed, correspondence matching is integral to disparity/depth recovery. Hence a quintessential way of categorizing stereo algorithms is the metric used to measure the affinity of two potentially matching pixels. The affinity of pixels is based on their pixel values which are a manifestation of how both cameras in the stereo rig react to light. Their responses to light are often not consistent, i.e. the pixel value of an object in one camera might be slightly different to that in the other. Hence, a more robust affinity test between two pixels (in separate cameras) will also consider the pixel values in their respective neighborhood. Given two image regions, a *matching cost* determines a real number that characterizes the degree to which the regions match. Generally, region-based matching cost functions are of three categories, namely: parametric, non-parametric and Mutual Information (MI) [25], parametric cost function being the most popular due to its computational efficiency.

Common parametric matching cost functions include: Sum of Absolute Differences (SAD), and Sum of Squared Differences (SSD) each with locally-scaled and zero-mean versions, Locally-scaled Sum of Absolute Differences (LSAD), Zero-mean Sum of Absolute Differences (ZSAD), Locally-scaled Sum of Squared Differences (LSSD) and Zero-mean Sum of Squared Differences (ZSSD). Another type of parametric matching-cost is Normalized-Cross Correlation (NCC), (with a zero-mean version - ZNCC) [25]. Each of the cost functions assumes an already rectified image pair with corresponding matching pixel only horizontally displaced in the other image. SAD is arguably the simplest of the region-based cost functions. SAD is calculated by taking the sum of the absolute difference of all intensity levels between the pixels within a neighborhood in the first image and those in a potentially matching neighborhood in the second image. The cost function can be mathematically described as follows:

$$C_{SAD}(p,d) = \sum_{q \in N_p} |I_L(\boldsymbol{q}) - I_R(\boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix})|, \tag{2.1}$$

where a corresponding search is made for pixel p with a vector position $\boldsymbol{p}$ on the left image plane - I_L; d denotes the number of pixel-shifts away from p in the horizontal line; and q denotes a pixel (with vector position $\boldsymbol{q}$ on the right image plane - I_R) within a neighborhood around p, called N_p. Note $\boldsymbol{q} - [d, 0]^T$ denotes the resulting vector position as a result of a horizontal shift of the vector $\boldsymbol{q}$ by $[d, 0]^T$. SSD is similar to SAD except that the differences are squared before summation within the region. This additional step means that it requires slightly more computation than SAD. Formally,

$$C_{SSD}(p,d) = \sum_{q \in N_p} \{I_L(\boldsymbol{q}) - I_R(\boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix})\}^2. \tag{2.2}$$

The locally-scaled variants of SAD and SSD attempt to compensate for gain bias by multiplying each pixel value in one of the two neighborhoods to be compared by the ratios of the mean intensity value of both regions. The equations are as follows.

$$C_{LSAD}(p,d) = \sum_{q \in N_p} |I_L(\boldsymbol{q}) - \frac{\overline{I_{N_p,R}}}{\overline{I_{N_p,L}}} I_R(\boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix})| \tag{2.3}$$

and

$$C_{LSSD}(p,d) = \sum_{q \in N_p} \left\{ I_L(\boldsymbol{q}) - \frac{\overline{I_{N_p,R}}}{\overline{I_{N_p,L}}} I_R(\boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix}) \right\}^2, \tag{2.4}$$

where the overbar denotes the mean.

NCC [26] is the most computationally expensive of the parametric cost functions considered in this thesis. The NCC matching cost is derived from cross-correlation which is effectively the integration of the product of two signals. These signals would have an amplitude distribution about the zero level. The NCC employs normalization first to ensure that the image intensity values (which are always positive) are distributed about the zero level. Formally,

$$C_{NCC}(p,d) = \frac{\sum_{q \in N_p} \left\{ I_L(\boldsymbol{q}) I_R(\boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix}) \right\}}{\sqrt{\sum_{q \in N_p} \{ I_L(\boldsymbol{q})^2 \} \sum_{q \in N_p} \left\{ I_R(\boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix})^2 \right\}}}. \tag{2.5}$$

The zero mean variants, ZSAD, ZSSD and ZNCC, also attempt to account for a constant gain bias radiometric difference. They achieve this by subtracting the intensity value of each pixel within the region of interest by its mean. Hence the transformation is as follows:

$$I_T(\boldsymbol{p}) = I(\boldsymbol{p}) - \overline{I}(\boldsymbol{p}), \tag{2.6}$$

where

$$\overline{I}(\boldsymbol{p}) = \frac{1}{|N_p|} \sum_{q \in N_p} I(\boldsymbol{q}). \tag{2.7}$$

This transformation is applied before the respective correspondence cost is carried out. There are other variants of parametric matching cost functions, for example, Maximum Normalized Cross-Correlation (MNCC) which is an approximation of the NCC with faster computation [27].

Non-parametric matching-costs are invariant to monotonic grey value changes. They rely solely on the relative intensity levels of pixels within the region. This allows them to tolerate a large class of local and global radiometric changes [28]. The Rank matching cost and Census cost [29] are two major types of non-parametric techniques. The Rank matching cost transforms the intensity level of each pixel to its intensity ranking within the neighborhood. This transformation is used as a correspondence match by computing the absolute difference. This is known to be sensitive to noise in textureless regions [25]. Formally, the Rank transform is formulated as:

$$C_{Rank}(p,d) = \sum_{q \in N_p} |T_{Rank}\Big\{I_L, \boldsymbol{q}\Big\} - T_{Rank}\Big\{I_R, \boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix}\Big\}|, \tag{2.8}$$

where the rank transform, T_{Rank} is defined as:

$$T_{Rank}\{I, \boldsymbol{q}\} = \sum_{r \in Nq} \mathbb{I}[I(\boldsymbol{r}) > I(\boldsymbol{q})]. \tag{2.9}$$

$\mathbb{I}[.]$ is an indicator function that returns 1 when the argument is true and 0 otherwise, and r are pixels around pixel q. The work of [30] uses the Rank matching cost to achieve improved disparity accuracy. Rank matching cost is particularly robust against brightness differences and image distortion [24], however, a consequence of the function is the tendency for ambiguity between sets of potentially matching costs. [31] presents the use of a Bayesian model to reduce the ambiguity that could exist. The Census cost also applies a comparison between the center pixel of the window region with the other pixels, however, provides a more granular comparison by translating it into a bit string. This is normally used in conjunction with the Hamming distance to compute the distance between the two bit strings describing the two regions whose affinity is to be evaluated as in:

$$T_{Census}\{I, \boldsymbol{q}\} = Bitstring_{r \in Nq}\Big\{\mathbb{I}[I(\boldsymbol{r}) > I(\boldsymbol{q})]\Big\}. \tag{2.10}$$

The actual distance is acquired as in

$$C_{Census}(p,d) = \sum_{q \in N_p} Hamming\{T_{Census}(I_L, \boldsymbol{q}), T_{Census}(I_R, \boldsymbol{q} - \begin{bmatrix} d \\ 0 \end{bmatrix})\}, \tag{2.11}$$

where $Hamming\{.\}$ is the bit-wise distance between two bit strings. A major downfall to the Census cost is that it often yields incorrect matches when there are repeated patterns in a region [32].

The final category of matching cost function is MI. Whilst parametric cost functions are solely sensitive to the magnitude of pixel intensity and non-parametric costs are sensitive to the local ordering of this pixel intensity within a neighborhood, Mutual information costs, model “a more complex relationship between the images in question” [33]. Statistically, MI measures the strength of association between two random variables. It conveys the number of instances in which two events are observed together in comparison to when they are not observed together. In terms of stereo image correspondence, the random variables are the pair of potentially matching pixels. Conventionally, computing MI for a stereo image pair requires an initial disparity. This disparity is used to warp one of the stereo pair so that corresponding pixels exists in the same spatial location in both images. Hence, disparity estimation using MI entails of proposing a disparity map that yields a minimal mutual information cost. Egnal [34] proposed the method for using mutual information for local stereo correspondence. At each pair of matching neighborhoods, the probability distribution over pixel values is computed individually and jointly by using a binned histogram. For instance, consider two potentially matching regions, N_L, and N_R, in the left and right image, I_L and I_R. First, a histogram of pixel values is established for each region. Using this histogram, the probability of a pixel ($x_L \in N_L$ or $x_R \in N_R$) having a pixel value can be estimated, $Pr\Big(I_L(x_L) = X\Big)$ and $Pr\Big(I_R(x_R) = Y\Big)$. In a similar manner, the joint probability of a pair of corresponding pixels, x_L, and x_R, in having a unique pair of pixel values can also be estimated. Then the mutual information of two potentially matching regions, $MI(N_L, N_R)$, can be computed as in

$$MI(X,Y) = \sum_{X,Y} Pr(X,Y) \log \frac{Pr(X,Y)}{Pr(X)Pr(Y)}, \tag{2.12}$$

where X and Y are the possible pixel values that the pixels in N_L and N_R can have respectively.

Another common variant of MI is Hierarchical Mutual Information (HMI) [35]. This uses a coarse-to-fine technique, by scaling down the images and then gradually scaling up. Starting with a randomly allocated disparity map, the images are displaced and the cost is computed. As previously discussed, an initial disparity image is needed to warp one of the stereo image pair so as to potentially register with the other. In a hierarchical approach, first, the stereo images $\{I_L, I_R\}$ are down-sampled by 2^f (e.g. 16) factors, $\{I_L^f, I_R^f\}$. The effect of this is a small disparity range, whereby a random sample of potential disparity solution will suffice to establish a match. This recovered disparity map solution, D_L^f is up-scaled to initiate the random disparity search for $\{I_L^{f-1}, I_R^{f-1}\}$. This process is repeated until the original scale is reached. The significance of this is that it reduces the computation time in comparison to the iterative computation involved in MI under full resolution. Experiments presented in [35] show that HMI produces matching qualities that are equal to MI.

2.1.2 Classification based on Cost Aggregation

Cost aggregation is applied to avoid basing affinity solely on a single pixel by including neighboring pixels. Consequently, the cost can be evaluated across a region of pixels (typically square window defined regions centered on the pixel of interest - Figure 2.2a). Cost aggregation approaches can be classified as either mask based or non-mask based. Non-masked based approaches solely aggregate based on the position of the contributing neighboring pixel without any weighted masking involved. A typical example of this is shifting window aggregation, where aggregation is selected from

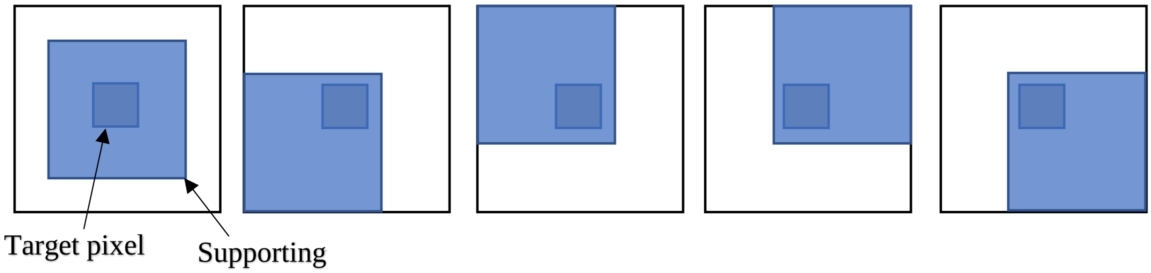


Figure 2.1 Multiple Window Cost Aggregation. This illustrates how multiple supporting windows regions can be established around a single target pixel.

a subset of the supporting window regions, see Figure 2.1. The selected subset is determined by the cost at each of the supporting windows, as implemented by [36]. A significant drawback is that the relative positioning of pixels is often not preserved, specifically in high gradient regions. Adaptive Window is another non-masked technique which was developed to address this [37]. This is a slightly different approach, where pseudo-supporting windows are used as in Figure 2.2b. Here, a subset of distinct neighboring windows is considered as opposed to overlapping windows in the case of the shifting window aggregation approach. Figure 2.2d illustrates the different potential shapes of aggregation region.

Similar to the shifting window approach, the support window with the best cost is selected. The adaptive supporting window method, implemented in [37], has been shown to provide superior results, particularly in high gradient regions. A common variant of the adaptive support window is one where the weighting is based on the similarity of the neighboring pixel to the target pixel. For instance in [38] a real-time GPU based approach assigns higher weightings to costs at pixels whose intensity values are closer to that of the target pixel.

Mask based aggregation methods are accompanied with a mask which determines the emphasis placed on the cost, resolved at a neighboring pixel. One of the more common approaches is weight contribution based on the proximity to the target pixel as in Figure 2.2c. A more complex masked based approach applies a form of low pass filter on the actual region and uses the response to weight the contribution from neighboring windows.

2.1.3 Classification based on Disparity Computation

Here the different disparities are categorized based on how the resulting cost is used in determining the corresponding point. These are of two kinds, either locally based or globally based. Most stereo-matching algorithms are locally based. Here prediction of correspondence is solely calculated using the affinity of potentially matching corresponding pair points. This is often referred to as the “winner takes all” approach, as the point with the highest affinity is simply chosen as the corresponding point and no other global cues (like smoothness) are considered [24]. This is in contrast to global optimization, that uses a global prior in conjunction with a locally applied matching cost to determine the best correspondence. This is often formulated in the form of energy minimization [39–41]. To reduce the complexity and to regularize the minimization task, the smoothness constraint is often applied, whilst special consideration is made for highly discontinuous regions. A prominent member of this class of disparity optimization is belief propagation [42]. Here the stereo-matching problem is formulated as three coupled Markov random fields namely: the smooth disparity field; absence (or presence) of depth discontinuity and indicator of occluded regions. The Loopy Belief propagation inference algorithm [43] is used to approximate the posterior probability for stereo matching. An attractive feature of these approaches is that they provide holistic solutions to the pixel-based disparity. However, this is only efficient and able

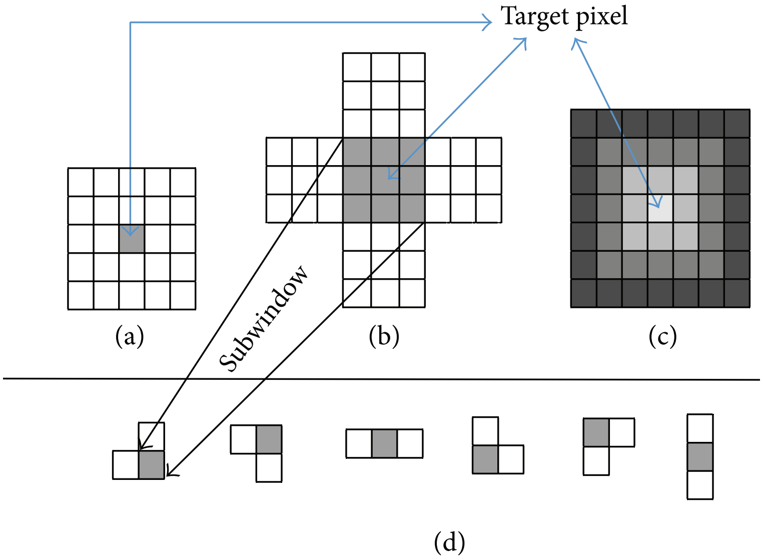


Figure 2.2 Cost Aggregation adapted from [24]. (a) Cost aggregation is typically evaluated across a square window defined region, centered on the pixel of interest. (b) Subsets of distinct neighboring windows are considered in the case of the shifting window aggregation approach. (c) A weight contribution based on the proximity to the target pixel. (d) Illustrates the different potential shapes of aggregation region.

to produce real-time solutions with low-resolution images where the potential levels of disparity are relatively small.

2.1.4 Classification based on Refinement

The refinement stage is a post-processing step to reduce the level of noise and artifacts in resolved disparity, typically involving the resolution of inconsistencies and closing of holes in the recovered disparity. These are achieved often by detecting pixels whose computed disparity differ from that at neighboring pixels. Occluded pixels detected by testing the consistency in both directions (left to right, and right to left) are often replaced with the maximum disparity [44].

2.1.5 Comment

Based on the above literature, stereo matching is a significant sub-field in computer vision and a wide variety of approaches have been proposed over the years to improve it. A major drawback in all of these approaches is that they often require experimenting with the approach to establish which one works best for different scenarios, and this is often done by hand. A more efficient approach will be to present a machine learning based method where the optimal affinity criteria can be achieved based on the dataset that sufficiently describes the context of intent. For instance, if the context of interest is hand depth estimation (as is the case in this thesis), a dataset of hand based images should be used to establish the affinity criteria. This thesis argues for the use of machine learning to learn depth estimation in the specific context of hand stereo imaging.

2.2 Hand Pose Estimation

As discussed in the Chapter 1, there are two main types of inputs used in hand pose estimation namely: RGB monocular cameras [12, 45–48] and active RGBD sensors [5, 9, 14, 49–52]. There are fewer methods in the literature that attempt to resolve for pose from stereo capture. This section discusses literature on hand pose estimation, specifically focusing on monocular RGB and RGBD based approaches. Stereo-based hand pose estimation will be discussed in the following section.

Hand pose estimation has been widely studied and the field has vastly developed over the years. This is particularly evident from the body of work described in the review papers of Pavlovic et al. and Supancic et al., [53] and [54] respectively, which were published only 18 years apart (1997 and 2015, respectively). The works presented by Pavlovic et al. in [53] overwhelmingly approach the problem from the perspective of RGB monocular cameras and are largely focused on extracting robust features that help discriminates an observed hand pose instance from a pool of potential poses. There is also a trend of establishing analytical prior over possible hand pose/gesture chase (e.g establishing optimal constraints on joint angle movements [55]). The more recent works presented by Supancic et al. in [54] are more RGBD based and more data-driven. Less emphasis is placed on the robustness of extracted to focus on more robust inference model and a larger quantity of data.

The recent surge in hand pose estimation interest is largely owing to the development of commercially available RGBD active depth sensors [56]. Estimating hand articulation with vision-based methods typically involves three components, including: the model used to mathematically represent the relationship between visual data and articulation state; the learning algorithm; and the inference algorithm that makes a prediction based on new visual data [57]. Most pose recovery frameworks can be differentiated based on the model type, which is either *discriminative* or *generative*. Generative techniques model the probability of the observed image, $\boldsymbol{x}$, given the state of the hand articulation, $\boldsymbol{w}$,

$$Pr(\boldsymbol{x}|\boldsymbol{w},\boldsymbol{\theta}) \tag{2.13}$$

where $\boldsymbol{\theta}$ represents the model parameters. In contrast, discriminative techniques model the probability of the state of the hand articulation given the observed depth data.

$$Pr(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{\theta}) \tag{2.14}$$

again $\boldsymbol{\theta}$ represents the model parameters. The learning and its complementary inference algorithm are generally categorized into purely regression-based or classification-based. Robust and accurate hand pose estimation requires discerning the 3D spatial joint locations (a continuous quantity) of the different consisting joints. This inherently makes it a regression problem, however, recent techniques have employed a combination of classification and regression learning algorithms, as in [58–60]. In [58], Tang et al. present a semi-supervised transductive model to enrich the expensive and limited real depth data with synthetic data using a Semi-supervised Transductive Regression (STR) variant of random forest. The approach involves viewpoint classification at top levels of the STR trees, followed by classification (clustering) of pixels into hand joints at the mid levels before finally regressing for high confidence voting of spatial hand joint location at the lower levels. It combines these three tasks in a single quality function (replacing the Information Gain) by embedding a switch into the quality function. The switch determines which task (classification, clustering or regression) the quality function is biased toward at different depth of the trees. In [59], Shotton et al. applied conventional Random forest to classify pixel in human body pose scene into different joints, before applying local mode-finding mean-shift algorithm to regress for the 3D location of joints. Keskin et al. apply a similar approach to hand estimation from depth in [60].

2.2.1 Generative Models

In a generative model-based technique, a hypothesis of visual data of the hand is generated using computer graphics, often using an articulated rendered 3D hand model. The main challenges with generative approaches are to establish the cost function that best represents the similarity of the hypothesized visual data to the perceived visual data and to implement effective optimisation techniques. Hence, the aim of the algorithm reduces to optimising for the articulation parameters that generate the synthetic scene to match the observed data. A positive consequence of using a 3D hand model is that potential ambiguity within hand poses observed from 2D captures can be resolved [45, 48].

Prior to the advent of RGBD sensors, monocular RGB images were the prominent input to hand pose estimation. These techniques faced several challenges which largely stems from the ambiguity in a 2D render of a 3D object that is inherent to monocular cameras [12, 45, 48, 61, 62]. More specifically, multiple 3D hand poses could yield the same 2D image when projected on an image plane. This highlights how ill-posed the problem is, as it does not have a unique solution. An obvious example of this is in the context of self-occlusion which is particularly prominent in hand poses [63]. This technique also suffers from a lack of generalization across hands of different types of subjects (i.e. skin tone, hand shape etc.) [64]. A common trend in the approach to addressing these problems is by modeling hand poses. The works presented by Stenger et al. [61] and Wu et al. [62] are two of the earlier proposals to address modeling human hand poses. They established the different degrees of freedom that each joint of the fingers exhibits and incorporating inverse kinematics to the optimization phase. Later work like [12, 48] present generative models of hand poses and the potential background in 2D. [12] presented the optimization of articulation as well as texture and scale of the hand to track the pose of the hand in the scene. In this work, de la Gorce et al. establish a synthetic hand scene comprising of a hand pose, texture and illuminant parameters and dynamically optimize these parameters by minimizing an energy function. This energy function describes the affinity between the synthesized scene and an RGB observed scene. A quasi-newton based optimizer is used to minimize the objective function in an efficient manner. In their earlier work, [48], de la Gorce et al. define the hand as an articulated model with pre-established kinematic constraints. To evaluate affinity, the corresponding hand silhouette projection with the observed hand is used. Similarly to [12], the hand pose is iteratively refined to minimize the affinity between projected silhouette of synthesized hand model hand and that in the observed RGB scene. The work in [45] introduces the use of hand pose capture gloves to collect a robust dataset of hand poses which allows it to establish a hand pose prior that is used in a Monte Carlo framework for local and global hand motion optimization.

The introduction of commercially available and reasonably cheap depth sensor cameras has caused a paradigm shift in the type of proposed techniques to address hand pose estimation. This is largely due to the fact that issues such as lighting variance, lack of shape information, 3D to 2D ambiguity, background clutter etc. were inherently improved with the depth input data [65]. This allowed researchers to focus on addressing pose estimation in a robust manner. Hence, a large proportion of recent literature in hand pose estimation has been proposed in the context of depth based inputs [51, 52, 66, 67]. A very prominent framework used in generative models is Particle Swarm Optimization (PSO) as in [14, 15, 63]. In these approaches, hand pose estimation and tracking are demonstrated in hand-object and hand-hand interacting scenarios by optimising for the parameters that yield the rendered depth that best matches the observed depth. Consequently, the performance relies on the availability of a powerful GPU to render potential pose solutions in real time. Oikonomidis et al. [51] were able to improve on their previous work, [63], by reducing optimization time of their framework by over a factor of four for each hand. It achieves this by proposing a quasi-sampling variant of the PSO framework. The work in [66] proposes the use of gradient as well as stochastic based optimization to achieve optimizations that are initialized by fingertip detections. [52] reverts back to the use of hand motion mechanics to establish constraints over hand pose inference.

Even with the introduction of depth sensors, a prominent issue with generative models is the struggle to generalize across multiple hand sizes and shapes. The key challenge of ensuring that the generated hand render (based on a given pose) still has a strong affinity with the observed hand (of the same pose) still remains, whether in a depth-based or RGB based input context. In the case of humans, the shape and size of hands vary, hence the task is split into either developing a more robust cost that is invariant to change in hand sizes or developing a more generalizable model (or modeling framework). The majority of the work in the literature appears to take the later approach [12, 48, 52, 68, 69]. One trend in approaches to having a more generalizable model is applying scale calibration to the subject's hand before testing, this allows for adjusting for the appropriate size 3D model [12, 45, 46, 48]. The work of [14, 15, 51] take a similar approach albeit manually achieved. The challenge of establishing generalizable hand models has prompted the research into a standardized framework for calibrating a model for different subjects. The work presented in [70–72] are a prominent few. In [71], Taylor et al. present an approach of approximating "dense non-rigid shape and deformation" from depth acquired from depth sensors. The task here is to recover a rigged mesh model that is specific to a particular user. A cost measure was proposed that quantifies the affinity between the deformable hand template and 15 frames depth capture from the subject's hand. The approach defines an initial 3D hand model mesh (a collection of 3D vertex points), denoted as $V_{template}$, and a subject-specific mesh as V_{core}. The proposed technique iteratively resolves for a mesh (a each frame, i), denoted $V^{i}_{instance}$, that is most consistent with the surface of the depth image of each captured frame instance whilst maintaining two constraints. First, that a skinned instance of $P_i(V_{core})$ has a matching morphology to each $V^{i}_{instance}$ and secondly, that V_{core} has a matching morphology to $V_{template}$. Here, $P_i()$ represents skinning function applied to V_{core} at each frame, i. The consequence of this is that the morphology of V_{core} will maintain a general structure of a conventional hand (based on $V_{template}$), whilst still matching closely with the shape of the subject's hand that is in the depth captures. Khamis et al. improved on their prior work, [71], by eliminating the need for the long sequence of hand captures in [70]. Here a parametric approach was proposed that uses established bases from datasets of captured sequences from different subjects with a varying range of hand shapes and sizes. In [72], Asad et al. recovered hand joints from RGB and depth frontal capture of the subject's hands using a Naive Bayes model. The extracting of these hand joints were based on visual cues (such as creases). A pre-designed hand model can then be warped such that the joint location from this model registers with the detected joint position from the frontal scan under rigid and non-rigid registration. The non-rigid registration is achieved by minimally skinning the pre-designed hand model whilst thin-plate-spline deformation (proposed in [73]) is applied to achieve rigid registration.

Generative model-based techniques are very flexible in that they can potentially resolve any pose without the need of establishing a dataset that covers all possible hand poses. However, the very high dimensionality of the search space poses a critical challenge, and as a result, methods often exploit predicted pose and measured visual data from a previous frame in initializing the optimization. Hence rapid and abrupt pose changes can lead to errors. A more severe consequence of this is that errors can accumulate (also known as drift) as demonstrated in [58]. However, it can be argued that increased frame rates for capture will reduce this issue. Still, an increase in frame rate will require an efficient approach (and powerful computing systems) to process these frames in sync with the data capture.

2.2.2 Discriminative Models

The success of [59] with the Microsoft Kinect has made discriminative techniques a prevalent model. The model learned constitutes the probability distribution of the pose of the hand whose parameters are dependent on the observed depth image data. A typical example of this is in the work of Keskin et al. and Fanello et al.([60] and [74] respectively) where the framework described in the work of Shotton et al, [59], is reapplied to hand pose estimation. These techniques aim to first establish the spatial position of each joint of the hand by classifying each pixel before computing the general pose of the hand. Discriminative techniques tend to use independent frame based prediction, i.e. prediction of hand articulation can solely be based on observations in the current frame. This is an attractive feature as cumulative errors from previous frames do not occur, inherently avoiding the drifting problem attributed to generative approach. However, this, of course, means not exploiting prior knowledge of the previous state of the hand articulation. A major drawback to these techniques is that they often require a very large labeled data set to capture the range of variation of hand poses and orientations relative to the camera. This is an expensive procedure and hence other means of addressing this have been recently explored, for instance, semi-supervised transductive learning as in [58].

A large body of recent discriminative model-based techniques for hand pose recovery is based on Random Forest [8, 20, 58, 60, 75]. This trend is largely inspired by the similar use of Random Forest to estimate human body pose using RGBD sensors [59, 65]. The work presented in [46] is one of the earlier examples of a discriminative model in hand pose recovery. Here a colored hand glove is worn and observed with a monocular RGB camera then pixels of observed 2D images are classified based on their color to yield a tiny image (see Figure 2.3). A tiny image is a low-resolution image whose pixels can only have a limited number of possible intensity values.

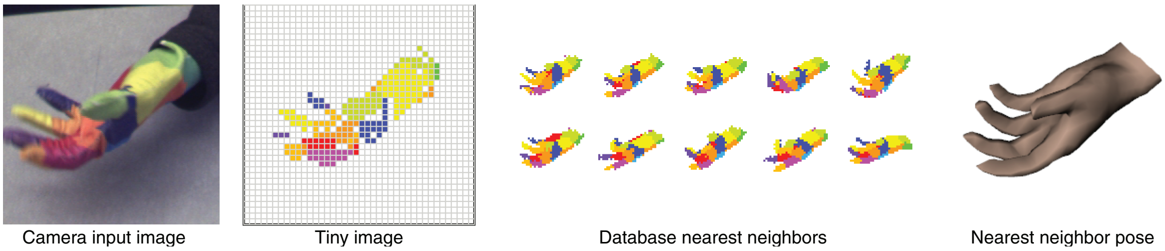


Figure 2.3 Colored glove approach proposed in [46]. Given an observed colored glove, pixels are classified based on color to acquire the "tiny image". A search is made from a large database of synthetic tiny images for the optimal pose.

A metric of affinity between different tiny images is established and used to search from a database of synthetically generated tiny images for the closest match. This was also one of the earlier use of synthetically generated images in real hand pose detection. The work in [60] also uses synthetic data. Specifically, it uses a dataset of synthetic hand depth images to train a Random Forest for real depth based pose recovery. [20] takes a similar approach to estimating hand pose and shape classification. It addresses the problem of hand shape classification with Random Forest and then proposes an expert multi-layered Random Forest framework to resolve for hand pose for a determined shape of the hand. As well as the Random Forest learner, [20, 60] also adopted the vector-based feature in [59]. This is a weak feature generally computed as

$$f_\theta(d_I, \boldsymbol{x}) = d_I\Big(\boldsymbol{x} + \frac{\boldsymbol{u}}{d_I(\boldsymbol{x})}\Big) - d_I\Big(\boldsymbol{x} + \frac{\boldsymbol{v}}{d_I(\boldsymbol{x})}\Big), \tag{2.15}$$

where $\boldsymbol{x}$ represents the 2D location of the pixel of interest, $\boldsymbol{u}$ and $\boldsymbol{v}$ are randomly oriented vectors representing a direction (as illustrated in Figure 2.4) and d_I is the depth image acquired by the RGBD sensor.

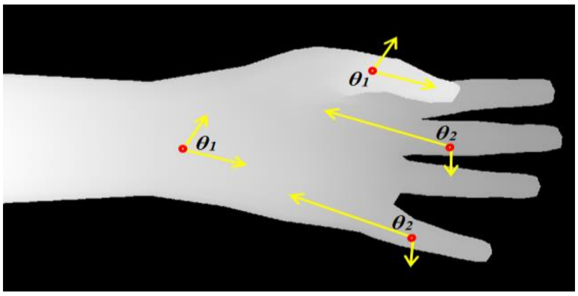


Figure 2.4 Illustration of the vector feature as a weak joint classifier. Feature θ_2 will discriminate between middle finger and the little (pinky) finger. Feature θ_1 will discriminate between a finger and the palm.

The idea here is that the feature θ_i is described by two arbitrary offset vectors ($\boldsymbol{u}_i$ and $\boldsymbol{v}_i$). Computing the feature valuef_θ at a given pixel location, $\boldsymbol{x}$ requires evaluating the difference in depth values at two locations (on the depth image of the scene) determined by the offset of $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ away from $\boldsymbol{x}$. This consequence of this is illustrated in Figure 2.4 above, where the offset vectors are represented by the yellow arrows whilst the red dot represents unique pixel locations. Observe the figure and notice how the depth feature computed at the thumb will be large in comparison to that taken at the palm/wrist region. This allows it to weakly discern fingers from the palm or wrist. With similar logic, feature θ_2 is able to disambiguate the middle finger and the little finger. This vector feature, which was originally adapted from the work presented in [76], has become very prominent in the discriminative pose recovery literature. This is because of its computational efficiency. The work in [7, 8] are a few examples. More recent work like [75] have extended upon this to make it invariant to a 2D plane rotation. This manifests as

$$f_\theta(d_I, \boldsymbol{x}) = d_I\Big(\boldsymbol{x} + \frac{\boldsymbol{R}_\alpha^x \boldsymbol{u}}{d_I(\boldsymbol{x})}\Big) - d_I\Big(\boldsymbol{x} + \frac{\boldsymbol{R}_\alpha^x \boldsymbol{v}}{d_I(\boldsymbol{x})}\Big), \tag{2.16}$$

where $\boldsymbol{R}_\alpha^x$ is a matrix that rotates its coefficient vector by an angle of α about the 2D point $\boldsymbol{x}$ on the image plane. Note α is initially regressed in a pre-step.

In the work presented in [8], instead of building a Random Forest based on the entropy of hand pixels as was the case in prior literature, a latent regression forest was proposed, that rather attempts to partition an image into different joints in a coarse-to-fine hierarchical manner. The vector-based features still remain the splitting feature at decision trees' nodes. [75] on the other hand, first estimates hand orientation before resolving for the pose. More recent approaches like [7, 77] have focused on improving upon [20, 60]. In [7], a cascade pose regression framework (introduced in [78]) is used to iteratively improve the proposed pose solution to an observed depth image. This is presented in the form of an update function, that computes an improved pose solution given the observed depth image and a currently proposed pose. Similar to [8], [7] also exploits the hierarchical structure of the hand, noting that parent joints (such as the palm) are more stable in comparison to child joints (like the fingertips) hence it proposes to sequentially regress for the spatial position of hand joints. For instance, the regression of the palm's spatial position will inform the position of the proximal joints and so on. [77] presents a more nuanced approach to pixel-based classification in a four stage pipeline approach. From pixel classification using Random Forest, hand part pixel region is segmented into super-pixels (to reduce the complexity of the Markov Random Field (MRF) that follows). The MRF is applied to re-infer pixel classification under the consideration of depth pixel smoothness etc. Finally, hand joint estimates are determined based on the centroid of labeled pixels.

The reinvigoration of neural networks in recent literature (due to the recent capability of processing and storing larger datasets) has brought Convolutional Neural Networks (CNN) to the forefront of hand pose estimation. Similar to the Random

Forest pixel classification approach, CNN based hand pose recovery was inspired by initial work done on human pose estimation, for instance, [79–81]. Specifically, [80, 81] present the advantage of using heat maps (that indicates the likelihood of a body joint being at a spatial location) as the target space for CNNs rather than using the raw spatial joint location. Here a mean squared error distance between forward-pass output heat map and the groundtruth heat map is minimized. This has inspired their other work [9], where a similar approach has been applied to the hand. The work of [82] identified that the 2D heat map used by Thompsom et al. only utilizes 2D information of joints and that the depth dimension was not fully exploited. More specifically, the argument presented was that prior approaches focused on solely solving 2D inference and once the 2D position of a joint is identified in the 2D image plane, the input depth image is used to get the depth component. This was identified as an issue in scenarios where minor errors in the regressed 2D plane can result in allocating the wrong depth to a joint, for instance at finger edges in a depth map. Instead, [82] proposes a 3D point cloud projection of the hand into three orthogonal planes and simultaneously feeds this into a three separate CNNs whose outputs are recombined and regressed for the 3D spatial position of hand joints. Note, a heat map is still used as a target in this implementation. The work in [83] experimented with a different CNN architecture for direct regressive mapping from the depth image to a 1D vector that consisted of the 3D spatial location of all joints of the hand.

2.2.3 Hybrid Generative-Discriminative Models

The pros and cons of using either generative or discriminative approach to modeling the hand pose recovery problem have prompted methods that attempt to leverage the advantages of both approaches. The work in [5, 9, 49, 50, 84] are a few examples of this. These approaches address the problem of cumulative error during tracking, inherent to generative methods, by re-initializing generative tracking based framework with a discriminatingly resolved pose solution.

The previously discussed work in [75] also exhibits traits of a hybrid approach. Here, a three-step framework was proposed that initially estimates the hand plane orientation and location in 3D. Resolved orientation is used to reorientate the hand to a frontal planar pose. The second step involves using vector-based features to propose 3D hand pose candidates using the Hough forest model. The final step verifies the predicted pose by rendering a synthetic hand pose based on this proposal and comparing with observed depth in a generative model-based approach. Consequently, a discriminative model is used to initialize different iterations of the generative model. The work of Toby et al. [50] takes a more corrective approach in that tracking failure added by the generative facet of their approach is corrected for by a discriminative model. This allows their method to achieve robust and flexible pose recovery. A two-layer re-initialization Random Forest is used, where the first layer is trained to predict the general rotation of the hand quantized into 128 rotation bins. In the second layer, three predictors are trained per bin, each trained to regress for a finer global rotation of the hand pose, offset of the hand, and a rough estimate of the hand pose. To regress for the finer pose, a generative model based on Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) is used to regress for the pose that yields a similar rendered image to the observed image. This regressor uses information from the coarse pose prediction from the two-layered Random Forest as well as resolved pose from the previous frame. The work in [84] presents a very similar approach, where a regressor is trained to propose initial potential solutions to pose. These proposals are used to initiate kinematic parameters of a hand model and evaluate them for affinity with observed data.

Hybrid model-based methods are more recent and generally entail of the use of a discriminative approach for initializing or correcting generative model based regressors. In the latter part of this thesis, a different approach is presented that uses a discriminative framework to evaluate for pose proposals from a generative model.

2.3 Stereo based Hand Pose Estimation

This section focusses on the limited literature of stereo based approaches to hand pose estimation. The recovery of hand depth provides unique challenges that differentiate the problem from depth recovery of arbitrary scenes. Unlike generic scene depth estimation there is significantly less texture, which makes stereo-matching substantially more challenging. There is also a high tendency of self-occlusion which manifests in changes in depth that might not reflect in a change in texture. For example, the occlusion of a finger on the palm will yield a change in depth but the color and the texture of the region of occlusion might remain largely unchanged as the color of the skin might be consistent (whether on the finger or on the palm region).

Unlike active depth camera based input, less work has been performed on stereo-based passive camera input for hand pose/gesture recognition. Techniques that are proposed to address stereo based hand pose estimation are largely grouped into two main categories, namely: depth map and non-depth map based.

2.3.1 Depth based Stereo Hand Pose Recovery

Depth map based techniques assume that the mapping between the stereo input and hand pose is strongly based on disparity information, which is a hidden variable. This is largely influenced by the recent success in robust hand tracking and pose estimation from depth images. These techniques attempt to recover a dense or at least a semi-dense, depth image before applying state-of-the-art depth based pose estimation. An example of this is [64], where a robust technique that focuses on depth recovery of hand pose scene is presented with the aim of later using it for hand pose estimation. The method utilizes an Adaptive Gaussian Mixture Model GMM segmentation to localize the hand skin region before recovering disparity based on stereo matches. Using the estimated hand skin region, it refines the disparity image recovered by constraining the disparity from proposed stereo matches. Finally, hand segmentation is further applied to the final disparity and uses [7] to track hand poses based on the recovered disparity image. A key drawback of this approach is that it assumes that the stereo algorithm will recover disparity/depth with the same consistency and accuracy. This is not always the case particularly with a low-quality stereo camera like the one used in this thesis. An erroneous disparity recovery will yield a wrong pose.

2.3.2 Non-Depth based Stereo Hand Pose Recovery

Non-depth based approaches, while still exploiting parallax information, do not attempt to explicitly extract a depth map of the scene. This is typified by the approach presented in [85]. Here a generative hand model approach is used to optimize the appropriate hand pose that yields stereo color consistency between the two cameras. Like most model-driven approaches in hand pose recovery, it does not require the tedious procedure of establishing a robust dataset. However, the approach does require an explicit definition of the anatomical size and hand pose constraint for the skinned model. Also, because of the method's temporal dependency, it is sensitive to the initialization of the pose. Another example is [86], where the pose estimation was preceded by first extracting the hand contour in both images in the stereo pair before matching points along the contour in one image to those in the other using dynamic time warping. This allows for the reconstruction of a 3-D contour of the hand, used to establish the hand contour tracking for subsequent finger tracking. Again, this approach is sensitive to the starting point selection to determine which pair of points on the contours serve as a seed to subsequent correspondence matching. Nonetheless, this only results in an aggregative tracking of the finger and pose, not providing a dense estimation of the spatial position of the other joints of the hand for a complete hand gesture/pose estimation.

2.4 Summary

A comprehensive review of literature relating to the task of hand pose recovery from stereo capture has been discussed. This included general stereo-matching literature; general hand pose estimation and then stereo based hand pose estimation. Stereo

Ref.	Model		Input Type			Trait					Method	Problem
	GM	DM	Dpt	MC	SC	GEN	MCL	CCL	GDC	RSH		
[14]	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	PSO	Single hand
[52]	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	3D skeletal tracking with physical constraints	Single hand
[12]	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	Quasi-Newton	Single hand
[48]	✓	✗	✗	✗	✓	✗	✓	✗	✗	✓	Gradient descent + Particle filter	Single hand
[45]	✓	✗	✗	✗	✓	✗	✓	✓	✗	✓	Sequential Monte Carlo	Single hand
[87]	✗	✓	✗	✗	✓	✗	✗	✗	✓	✗	Nearest neighbour based on chamfer distance of edge images	Single hand
[68]	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	Nearest neighbour based on HOG features	Single hand
[69]	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	Gradient descent on SAG Models	Single hand
[88]	✓	✗	✗	✓	✗	✗	✗	✓	✗	✓	PSO	Single hand
[89]	✗	✓	✗	✓	✗	✗	✗	✗	✓	✓	Specialized mapping architecture	Single hand
[20]	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	Multi-layered Random Forest	Single hand

[5]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	CNN as a hand depth synthesizer + updater	Single hand
[75]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	Hough Forest + 3D hand model fitting	Single hand
[7]	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	Sequential regression based on hierarchical hand structure	Single hand
[58]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	Transductive Regressive Random Forest	Single hand
[60]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	Random Forest with vector feature	Single hand
[8]	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	Latent Random Forest on hierarchical segmentation	Single hand
[49]	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	Levenberg method	Single hand
[50]	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	Multi-layered Random Forest + PSO	Single hand
[9]	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	CNN + joint location heat map target	Single hand
[66]	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	ICP + PSO	Single hand
[90]	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	PSO + Ensemble of collaborative trackers	Single Hand + Object
[91]	✗	✓	✓	✗	✗	✓	✗	✓	✓	✓	Linear SVM	Single Hand + Object

[92]	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	ICP + GMM	Single Hand + Object
[15]	✓	✗	✗	✓	✗	✗	✗	✓	✗	✓	PSO	Single Hand + Object
[18]	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	Nearest neighbour	Single Hand + Object
[63]	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	PSO	Two Hands
[51]	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	PSO + Evolutionary random search	Two Hands
[67]	✓	✓	✓	✗	✗	✗	✓	✓	✗	✓	SVM + Gradient ascent optimizer	Two Hands
[93]	✓	✓	✗	✓	✗	✗	✗	✓	✓	✗	Optimizing pose based on re-projected salient points	Two Hands

Table 2.1 An overview of hand pose recovery. Key: **GM** - Generative Model, **DM** - Discriminative Model, **Dpt** - Depth based input, **MC** - Multi-Camera input(RGB), **MC** - Single-Camera input (RGB), **GEN** - Generalises to new subjects, **MCL** - Hand Model Calibration required, **CCL** - Camera Model Calibration required, **GDC** - Guided Dataset collection, **MCL** - Requires Specialized Hardware, **HOG** - Hstogram of Oriented Gradient, **PSO** - Particle Swarm Optimization, **SAG** - Sum of Anisotropic Gaussian, **ICP** - Iterative Closest Point, and **GMM** - Gaussian Mixture Model.

algorithms were classified and discussed in the context of the four main stages of stereo-matching, namely: Computing the matching cost; Cost Aggregation; Disparity selection and Disparity refinement. Hand pose estimation techniques, on the other hand, tend to differentiate based on the model (i.e. generative, discriminative or either). Generative model-based approach look to update 3D hand models in the search for the optimum affinity with observation, whilst discriminative approaches rely on large datasets and search for a member of the dataset that suits the observation. Stereo-based approaches are mainly classified based on how reliant the approach is on the depth hidden variable (i.e. whether it explicitly attempts to recover depth before pose estimation or not). A comprehensive comparison of recent related literature is presented in Table 2.1.

Nonetheless, a prominent point that cannot be ignored in this review is the gap in quantity and quality of research done on the task of hand pose recovery in the context of stereo-based input in comparison to the other two input types (monocular RGB and active depth sensor). This is the task this thesis aims to address: to explore the possibility of extracting robust pose estimation from a stereo input. The next three chapters present different stages to approaches this problem.

Chapter 3

Background

This chapter will introduce some preliminary concepts that provide the foundation to the concepts proposed in the later chapters of this thesis. Central to this dissertation is the concept of camera models, specifically multi-view camera models. The basis of camera models and multi-view geometry that pertains to this thesis is first introduced. These include the pinhole camera model, epipolar geometry, the concept of fundamental matrices and multi-camera registering. The second part of this chapter largely examines the machine learning aspect of the thesis. Basic concepts of inference and learning are introduced including regression vs classification, stochastic vs discriminative approaches, probability modeling, marginalization, and factorization, before exploring some of the machine learning frameworks that feature in the later chapters of the thesis. These includes Random Forests, Convolutional Neural Networks (CNN), Conditional Random Fields (CRF) and Markov Chain Monte Carlo (MCMC) samplers. The chapter concludes with a summary of the concepts that were introduced.

3.1 Camera Model and Multi-View Geometry

This section will introduce the pinhole camera model and multi-view geometry theory that forms the basis of the work presented in the later chapters of this thesis.

3.1.1 Pinhole Camera Model

The pinhole camera model is a mathematical model that describes the projection of a point in 3D coordinate space onto a 2D image plane of an *ideal* pinhole camera with an infinitely small aperture.

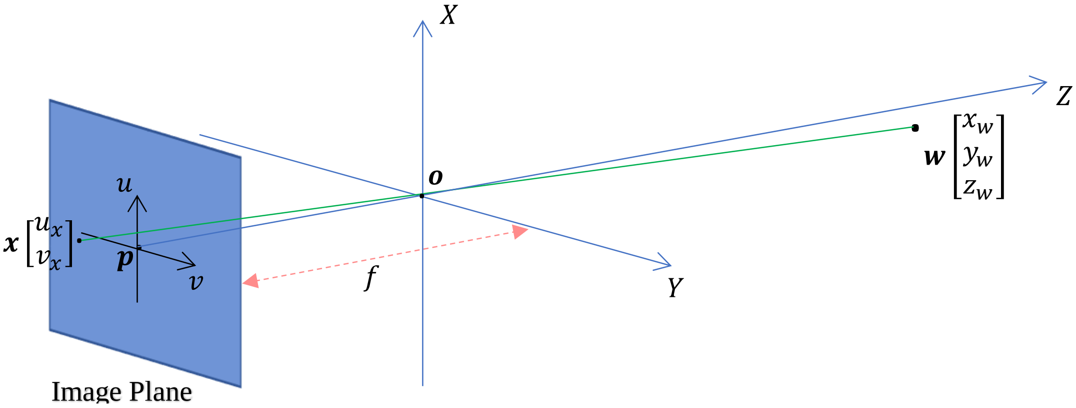

X
Z
Y
$\boldsymbol{o}$
$\boldsymbol{w} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$
u
v
$\boldsymbol{p}$
$\boldsymbol{x} \begin{bmatrix} u_x \\ v_x \end{bmatrix}$
f
Image Plane

Figure 3.1 Pinhole Camera Model. X-Y-Z represents the 3D camera coordinate system whilst u-v represents the 2D image plane coordinate system. Given the center of the camera coordinate system (optical center) $\boldsymbol{o}$, the principle point $\boldsymbol{p}$, and the focal length f, a 3D point $\boldsymbol{w}$ is projected onto the image plane at the 2D point $\boldsymbol{x}$.

Consider Figure 3.1, with a 3D orthogonal coordinate system with a center point, $\boldsymbol{o}$, that coincides with the camera aperture also referred to as the *optical center*. The the line that coincides with the Z-axis referred to as the *principal axis*, intersects with the *image plane* at the *principal point* $\boldsymbol{p}$. A real world 3D point, $\boldsymbol{w}$, is then projected onto the 2D point, $\boldsymbol{x} = [u_\text{x}, v_\text{x}]^\text{T}$, on the image plane. A key factor that affects the resulting projected point, $\boldsymbol{x}$, is the separation between the principle point and the optical center of the camera system, referred to as the *focal length*, f. Now, to explore how the relationship between $\boldsymbol{w}$ and $\boldsymbol{x}$ is affected by the parameters of the camera system, consider the X-Z plane view of the model as in Figure 3.2.

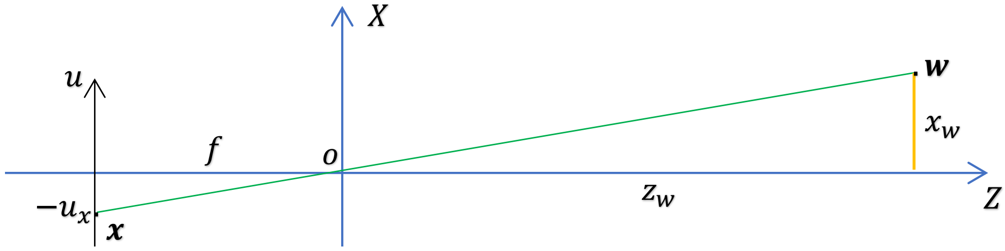


Figure 3.2 X-Z plane view of the pinhole camera model illustrated in Figure 3.1. In both figures the green line indicates the projection line of 3D point $\boldsymbol{w}$ onto the image plane. Observe how the ratio $x_w : z_w$ is equal to $u_x : f$.

From the figure, it can be deduced that,

$$u_x = -\frac{x_w f}{z_w}. \tag{3.1}$$

This can be replicated on the Y-Z plane to deduce that,

$$v_x = -\frac{y_w f}{z_w}. \tag{3.2}$$

To represent u_x and v_x in image pixel coordinates, the positioning of the photoreceptors on the image plane that capture light from the scene of interest is considered. To this end, ϕ is introduced. This is in effect the ratio between inter-pixel distance and a unit distance in the 3D world in question. Hence Eq. 3.1 and Eq. 3.2 can be represented as

$$\begin{aligned} u_x &= -\frac{x_w \phi_u f}{z_w} \\ v_x &= -\frac{y_w \phi_v f}{z_w}, \end{aligned} \tag{3.3}$$

where ϕ_u and ϕ_v are separate scaling factors in each dimension of the image plane. Lastly to account for the convention that the top-left corner of the image is considered the center of the image pixel plane system, u_x and v_x are offset with δ_u and δ_v

respectively. Finally the transformation can be fully represented as,

$$\begin{aligned} u_x &= -\frac{x_w \phi_u f}{z_w} + \delta_u \\ v_x &= -\frac{y_w \phi_v f}{z_w} + \delta_v. \end{aligned} \tag{3.4}$$

ϕ_u, ϕ_v, δ_u, δ_v, and f are referred to as the *intrinsic parameters* of a camera. In an ideal scenario like the one proposed in Figure 3.1, δ_u and δ_v can be interpreted as half of the width and height, respectively, of the image plane, assuming that the principal point is perfectly centered on the image plane. However, this is not the case in real systems due to imperfectly produced cameras.

Nonetheless, to mathematically model the projection of a camera, the challenge is reduced to solving for the camera's intrinsic parameters from a pair of known 3D points, $\{\mathbf{w}_i\}_{i=1}^{I}$ and their corresponding locations of projection on the image plane, $\{\mathbf{x}_i\}_{i=1}^{I}$. This yields a set of non-linearly related equations without a closed-form solution.

Homogeneous Coordinates

To address the non-linear problem above homogeneous coordinates are introduced. The non-linear 2D to 3D system of relationship can be represented in homogenous coordinate so that it becomes linear. This will allow for a closed-form solution to be formulated. To convert between the Cartesian to homogeneous coordinates simply requires of appending a 1 to the end of the coordinate vector as in from $\boldsymbol{x}_\mathrm{C} = [u_\mathrm{x},\, v_\mathrm{x}]^\mathrm{T}$ (in Cartesian coordinates) to $\boldsymbol{x}_\mathrm{H} = [u_\mathrm{x},\, v_\mathrm{x},\, 1]^\mathrm{T}$ (in homogeneous coordinates). Conversion from homogeneous to Cartesian coordinates (a process called dehomogenization) is achieved by dividing each element of the homogeneous coordinate vector with its last element, as in $\boldsymbol{x}_\mathrm{H} = [u_\mathrm{x},\, v_\mathrm{x},\, w_\mathrm{x}]^\mathrm{T}$ to $\boldsymbol{x}_\mathrm{C} = [u_\mathrm{x}/w_\mathrm{x},\, v_\mathrm{x}/w_\mathrm{x}]^\mathrm{T}$.

The consequence of this is that the mathematical relationship presented in Eq. 3.4 can be represented as,

$$\begin{bmatrix} -\phi_u f & 0 & \delta_u & 0 \\ 0 & -\phi_v f & \delta_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} -x_w \phi_u f + \delta_u z_w \\ -y_w \phi_v f + \delta_v z_w \\ z_w \end{bmatrix}_H \tag{3.5}$$

in the homogeneous coordinate system. The right hand size of Eq. 3.5 can be converted to Cartesian coordinate to resolve for u_x and v_x as in

$$\begin{bmatrix} -x_w \phi_u f + \delta_u z_w \\ -y_w \phi_v f + \delta_v z_w \\ z_w \end{bmatrix}_H = \begin{bmatrix} -x_w \phi_u f / z_w + \delta_u z_w / z_w \\ -y_w \phi_v f / z_w + \delta_v z_w / z_w \\ z_w / z_w \end{bmatrix}_C = \begin{bmatrix} u_x \\ v_x \\ 1 \end{bmatrix}. \tag{3.6}$$

Note that whilst the mapping from the 4D space to 3D is a linear one, the transformation from the homogeneous to the Cartesian coordinate system (dividing by the last element) that comes after is a non-linear one. Hence solutions derived from this system are not guaranteed to be a solution to Eq. 3.4, however, they can be a strong initializer for optimization techniques like Levenberg-Marquardt [94].

Just like points in space, lines posses homogeneous representations as well. A 2D line, $\boldsymbol{l}$ can be represented by the equation, $ax + by + c = 0$, where different values of a, b and c will yield different lines. The coefficients a, b and c will suffice to represent a line in the homogeneous domain as in, $\boldsymbol{l}_H = [a, b, c]^T$. Observe that for any 2D point, $\boldsymbol{x}_C = [\alpha, \beta]^T$ to exist on the line, l, then $a(\alpha) + b(\beta) + c = 0$. Similarly, in homogeneous representation,

$$\boldsymbol{x}_H{}^T \boldsymbol{l}_H = 0, \tag{3.7}$$

if and only if the point, $\boldsymbol{x}$ lies on the line l.

3.1.2 Stereo Camera Model and Epipolar Geometry

Stereovision is based on the physical concept of stereopsis. This specifies that given the view of a scene from two perspectives, the shift undergone by corresponding pixels in both images varies such that it is inversely proportional to the distance from the camera. Hence the problem of depth recovery given a pair of imaging devices is reduced to establishing a correspondence between both sets of pixels. The resulting image, whereby each pixel values indicate the separation between the pixel's location and the location of its correspondence, is referred to as *disparity*. To improve the speed and integrity of the correspondence search, the epipolar geometry of the stereo rig can be exploited. Consider Figure 1.1, where a stereo rig with two camera optical centers at $\boldsymbol{o}_L$ (left camera) and $\boldsymbol{o}_R$ (right camera), capture a 3D point $\boldsymbol{w}$ in an arbitrary world coordinate system. The plane on which $\boldsymbol{o}_L$, $\boldsymbol{o}_R$ and $\boldsymbol{w}$ exist is referred to the *epipolar plane*. If $\boldsymbol{w}$ is projected onto the left image plane at $\boldsymbol{x}_L$, then geometrically, the projection of $\boldsymbol{w}$ on to the right image plane, $\boldsymbol{x}_R$ is constrained to lie on the line $\boldsymbol{e}_R$. This line is referred to as the *epipolar line*.

As a result, the search for a corresponding point is constrained from a 2D search to a 1D search along the corresponding epipolar line. To further simplify the correspondence

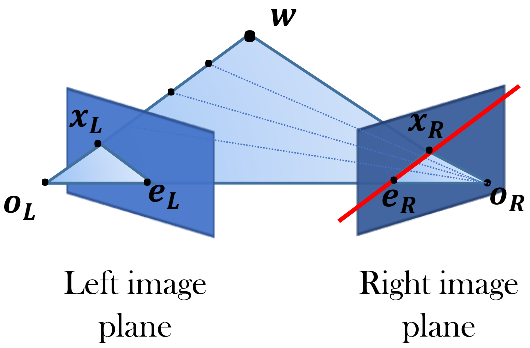


Figure 3.3 Illustrating epipolar geometry and the epipolar Line. The red line indicates the epipolar line in the right image, that corresponds to the image point $\boldsymbol{x}_L$ in the left. Observe how the points $\boldsymbol{o}_L$, $\boldsymbol{o}_R$ and $\boldsymbol{w}$ exists on a plane, this plane is referred to as the epipolar plane and the intersection of the epipolar plane with the image planes yields the epipolar lines

search algorithm two approaches could be taken. First, the captured stereo image could be warped such the set of corresponding matches exist on a single horizontal line. This is achieved mathematically through *stereo rectification.* In fact, most stereo-matching algorithms are generally preceded by a rectification step and most proposed stereo-matching techniques are presented assuming rectified images. An alternative approach will be to establish for a given point (in the first stereo image), the corresponding epipolar line (on the other stereo image) via the *fundamental matrix* and the *essential matrix.* Both of these scenarios require multi-view *camera calibration* .

Fundamental and Essential Matrix

The epipolar geometry of a stereo rig is embodied in the fundamental matrix, $\boldsymbol{F}$ or the essential matrix, $\boldsymbol{E}$. The essential matrix contains information on the translation and rotation between the image plane of the cameras in the stereo rig - referred to as *extrinsic parameters.* The fundamental matrix, like the essential matrix, contains the extrinsic information, however, it also contains information about the intrinsic parameters of both cameras. Hence, whilst the essential matrix encapsulates the mapping of a point from the physical camera space of one of the stereo camera to the other, the fundamental matrix encapsulates the mapping from the image plane of one of the stereo camera to the other.

Revisiting the stereo rig illustrated in Figure 3.3, recall that the coordinate of $\boldsymbol{w}$ is in the arbitrary world coordinate system; and that $\boldsymbol{x}_L$ and $\boldsymbol{x}_R$ are the projection of $\boldsymbol{w}$ on the left and right image planes. Let $\boldsymbol{p}_L$ and $\boldsymbol{p}_R$ be the 3D point coordinate of $\boldsymbol{w}$ in the left and right camera coordinate systems respectively. Then $\boldsymbol{p}_R = \boldsymbol{R}(\boldsymbol{p}_L - \boldsymbol{t})$, where $\boldsymbol{R}$ and $\boldsymbol{t}$ are is the rotation and translation from the right camera image plane to the left. Acknowledging that the vectors $\boldsymbol{p}_L$ and $\boldsymbol{t}$ (from the origin $\boldsymbol{o}_L$) both lie on the epipolar plane then the epipolar plane can be mathematically represented as $(\boldsymbol{p}_L - \boldsymbol{t})^T(\boldsymbol{t} \times \boldsymbol{p}_L) = 0$. The relationship between $\boldsymbol{p}_{\mathrm{L}}$ and $\boldsymbol{p}_{\mathrm{R}}$ can be substituted to get

$$(\boldsymbol{R}^T\boldsymbol{p}_R)^T(\boldsymbol{t} \times \boldsymbol{p}_L) = 0. \tag{3.8}$$

The cross product can be represented as

$$\boldsymbol{t} \times \boldsymbol{p}_L = \boldsymbol{S}\boldsymbol{p}_L \Rightarrow \boldsymbol{S} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \tag{3.9}$$

Consequently Eq. 3.8 can be represented as

$$\begin{aligned} \boldsymbol{p}_R^T \boldsymbol{E} \boldsymbol{p}_L &= 0, \\ \boldsymbol{E} &= \boldsymbol{R}\boldsymbol{S}. \end{aligned} \tag{3.10}$$

$\boldsymbol{E}$ is the essential matrix, as described above, encoding the extrinsic (rotation, translation) parameters. In contrast, the fundamental matrix, includes the intrinsic properties of both cameras. For the rest of this discussion, equations are presented in homogeneous coordinates. Let $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$ represent the intrinsic matrix of the left and right cameras respectively. Since $\boldsymbol{x}_L$ and $\boldsymbol{x}_R$ are 2D points on the left and right image planes where $\boldsymbol{w}$ is observed, then $\boldsymbol{p}_L = \boldsymbol{M}_L^{-1}\boldsymbol{x}_L$ and $\boldsymbol{p}_R = \boldsymbol{M}_R^{-1}\boldsymbol{x}_R$. Hence from Eq. 3.10 it can be inferred that

$$\boldsymbol{x}_R^T ({\boldsymbol{M}_R}^{-1})^T \boldsymbol{E} {\boldsymbol{M}_L}^{-1} \boldsymbol{x}_L = 0. \tag{3.11}$$

Lastly, this can be rewritten as

$$\boldsymbol{x}_R^T \boldsymbol{F} \boldsymbol{x}_L = 0 \tag{3.12}$$

where

$$\boldsymbol{F} = ({\boldsymbol{M}_R}^{-1})^T \boldsymbol{E} {\boldsymbol{M}_L}^{-1}. \tag{3.13}$$

Eq. 3.12 has a significant consequence in that two corresponding points on each stereo image can be mathematically related, given the knowledge of the fundamental matrix of the stereo rig. It should be noted that the rank deficient nature of the essential and the fundamental matrix implies that given a point on one of the image planes, and a knowledge of the fundamental matrix, there are several potential points (on the corresponding stereo image plane) that will conform to Eq. 3.12. Nonetheless, given Eq. 3.7, it can be deduced that the 2D line described in homogeneous representation as $\boldsymbol{F}\boldsymbol{x}_L$ must coincide with the point, $\boldsymbol{x}_R$. By definition, this is the epipolar line. Consequently, given a point on the left image plane, $\boldsymbol{x}_L$, its corresponding point, $\boldsymbol{x}_R$, must exist on the epipolar line, $\boldsymbol{l}_R$ which can be computed from the fundamental matrix as in

$$\boldsymbol{l}_R = \boldsymbol{F}\boldsymbol{x}_L. \tag{3.14}$$

Eq.3.14 is used in several instances of this dissertation to constrain the search for potential stereo correspondences to a 1D line.

3.1.3 Registering a Stereo Camera with an RGBD Sensor

A reoccurring scenario in this thesis is application of a machine learning technique to infer disparity information from a stereo capture with the aim of estimating a robust depth measure of the scene. This often requires ground-truth depth. In this thesis, the Microsoft Kinect depth sensor is used. To this end, it is required that there is a strong registration between one of the stereo images and the RGBD sensor image plane. To achieve this the fact that the color and the depth channel of these RGBD sensors are inherently very well registered is exploited. Hence the task of registering depth (captured from the RGBD sensor), with one of the stereo cameras is reduced to registering the color channel of the RGBD sensor with the stereo camera pair. Henceforth, the RGB channel of the RGBD sensor will be referred to as *RGBD-Color*. The registration of RGBD to stereo cameras is achieved by a stereo calibration of one of the stereo cameras to the RGBD colour camera using a checkerboard pattern. The following subsection will introduce the basics of stereo camera calibration.

Stereo Calibration

In a stereo setup, it is useful to know the extrinsic and the intrinsic parameters. The intrinsic parameter (also referred to as the projection matrix) defines the transformation from 3D camera space to the 2D image planes of the cameras, whilst the extrinsic parameter defines the relative transformation and rotation between both stereo cameras (as discussed previously). Whilst *camera calibration* is the process of establishing these parameters for a single camera, *stereo calibration* is a processing of establishing the geometric relation (extrinsic parameters) between the two cameras in a stereo rig.

An important pre-step in camera calibration is establishing a dataset consisting of 3D points and corresponding 2D location on the imaging plane as in $\{(\boldsymbol{x}_1, \boldsymbol{w}_1), ..., (\boldsymbol{x}_K, \boldsymbol{w}_K)\}$. This can be achieved by capturing images of a 3D object with prominent visual cues (whose 3D dimensions are known) with the camera(s) of interest and identifying the corresponding 2D location of these visual cues on the resulting images. This 3D object is referred to as a *calibration object.* A typical calibration object is a checkerboard of known dimension as shown Figure 3.5. Before delving into a discussion on stereo calibration, an important concept to review is *homography.*

Homography Planar homography is projective transform from one plane to another. This is significant in camera projection particularly when one considers the projection of a planar 3D object onto the image plane of a camera (see Figure 3.4). This transformation can be represented mathematically:

$$\boldsymbol{x} = \boldsymbol{H}\boldsymbol{w} \tag{3.15}$$

where

$$\boldsymbol{w} = [x, y, z, 1]^T, \tag{3.16}$$

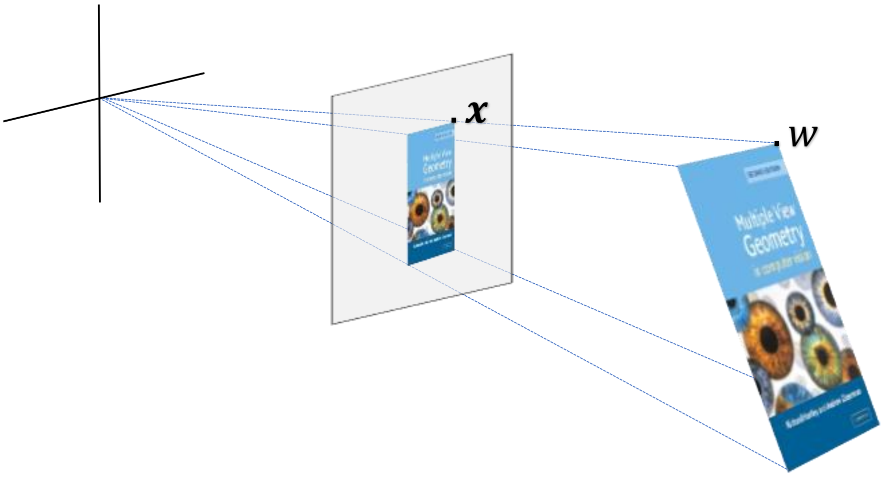


Figure 3.4 Projective Transform of a planar object, an example of a homography. The 2D point $\boldsymbol{w}$ on the book plane is projected to the point $\boldsymbol{x}$ on the camera image plane. This transformation can be represented with a homography.

the 3D real world coordinate, and

$$\boldsymbol{x} = [u, v, 1]^T, \tag{3.17}$$

the image plane location in homogeneous coordinate system. The elements of $\boldsymbol{H}$ can be established from a set $\{(\boldsymbol{x}_1, \boldsymbol{w}_1), ..., (\boldsymbol{x}_K, \boldsymbol{w}_K)\}$ using an iterative method such as Random Sample Consensus (RANSAC) [95].

Recall from that alternative a conventional camera model can be represented as

$$\boldsymbol{x} = \boldsymbol{M}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{w}, \tag{3.18}$$

where $\boldsymbol{M}$, $\boldsymbol{R}$ and $\boldsymbol{t}$ are the projection matrix; and the rotation matrix and translational vector from real-world coordinate to the camera coordinate system. Assuming the real world is defined by the planar object, such that a corner of this object is the origin of the real world coordinate system. Note that all points transformed from the planar object will possess a zero third component, due to the flat nature of a plane.

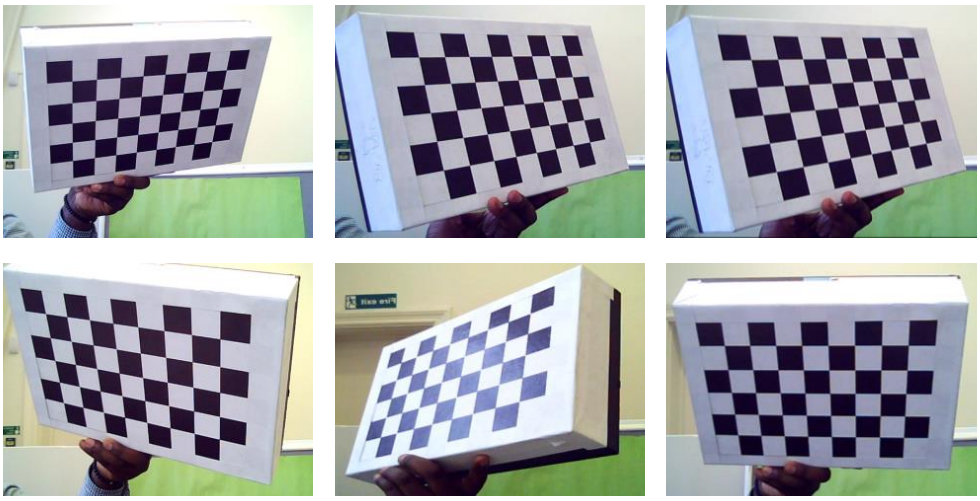


Figure 3.5 Sample captures of a calibration object (checkerboard pattern).

Consequently, Eq. 3.18 above be represented as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{M}[\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3|\boldsymbol{t}] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{M}[\boldsymbol{r}_1, \boldsymbol{r}_2|\boldsymbol{t}] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{3.19}$$

where $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$, are column vectors of $\boldsymbol{R}$. Note that a consequence of this is that the homography, $\boldsymbol{H}$ is a 3×3 matrix and

$$\boldsymbol{H} = \boldsymbol{M}[\boldsymbol{r}_1, \boldsymbol{r}_2|\boldsymbol{t}]. \tag{3.20}$$

Camera Calibration The camera calibration procedure consists of the capturing of the calibration object, positioned at different orientations relative to the camera of interest (see Figure 3.5). Each orientation yields a homography, $\boldsymbol{H}^i = [\boldsymbol{h}_1^i, \boldsymbol{h}_2^i, \boldsymbol{h}_3^i]$, which can be resolved from the establish correspondence based on the visual cues on the calibration object. $i \in 1, ..., K$, where there are K orientation captures. In the case of a checkerboard an corner edge detection algorithm is applied to establish the landmarks (square corners) on the checkerboard as illustrated in Figure 3.6. Consequently, from Eq. 3.20, it can be inferred that

$$\begin{aligned} \boldsymbol{h}_1^i &= \boldsymbol{M}\boldsymbol{r}_1^i \quad \text{or} \quad \boldsymbol{r}_1^i = \boldsymbol{M}^{-1}\boldsymbol{h}_1^i \\ \boldsymbol{h}_2^i &= \boldsymbol{M}\boldsymbol{r}_2^i \quad \text{or} \quad \boldsymbol{r}_2^i = \boldsymbol{M}^{-1}\boldsymbol{h}_2^i \\ \boldsymbol{h}_3^i &= \boldsymbol{M}\boldsymbol{t}^i \quad \text{or} \quad \boldsymbol{t}^i = \boldsymbol{M}^{-1}\boldsymbol{h}_3^i, \end{aligned} \tag{3.21}$$

where $\boldsymbol{r}_1^i$ and $\boldsymbol{r}_2^i$ are the first and second column vectors of the rotation matrix of the i^{th} orientation; and $\boldsymbol{t}^i$ is it's translation vector. Note there three component in Eq. 3.21 above, including: homography ($\boldsymbol{h}_1, \boldsymbol{h}_2$, and $\boldsymbol{h}_3$), the intrinsic matrix, $\boldsymbol{M}$, and the extrinsic parameters ($\boldsymbol{r}_1, \boldsymbol{r}_2$, and $\boldsymbol{t}$). The homography is known (resolved from using RANSAC above) and the aim to to determine the intrinsic and extrinsic. First to solve the extrinsic parameters, the intrinsic matrix is eliminated. To do this, the constraints that $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ should be orthonormal and their magnitude should be equal

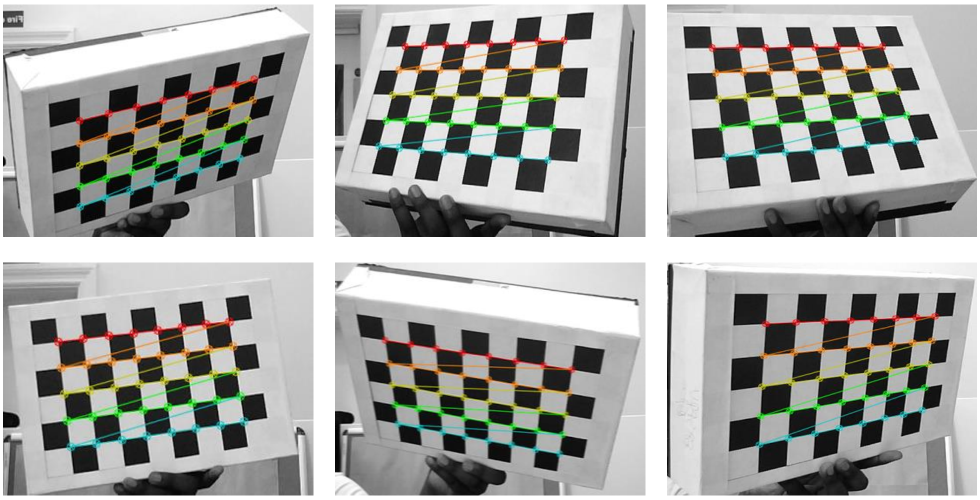


Figure 3.6 Illustration of checkerboard corner detection used to establish corresponding pairs of 2D image point and 3D spatial point.

are exploited, i.e. $\boldsymbol{r}_1^T\boldsymbol{r}_2 = 0$ and $||\boldsymbol{r}_1|| = ||\boldsymbol{r}_2||$. Consequently, from the first constraint,

$$\begin{aligned}
&(\boldsymbol{M}^{-1}\boldsymbol{h}_1^i)^T\boldsymbol{M}^{-1}\boldsymbol{h}_2^i = 0 \\
&(\boldsymbol{h}_1^i)^T(\boldsymbol{M}^{-1})^T\boldsymbol{M}^{-1}\boldsymbol{h}_2^i = 0.
\end{aligned} \tag{3.22}$$

and from the second constraint,

$$(\boldsymbol{h}_1^i)^T(\boldsymbol{M}^{-1})^T\boldsymbol{M}^{-1}\boldsymbol{h}_1^i = (\boldsymbol{h}_2^i)^T(\boldsymbol{M}^{-1})^T\boldsymbol{M}^{-1}\boldsymbol{h}_2^i. \tag{3.23}$$

This isolates the first unknown (intrinsic), eliminating the extrinsic. From Section 3.1.1, recall that

$$\boldsymbol{M} = \begin{bmatrix} f_v & 0 & \delta_v \\ 0 & f_u & \delta_u \\ 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

where f_v, f_u, δ_v and δ_u are the v-dimension focal length, u-dimension focal length, v-dimension pixel offset, and u-dimension pixel offset, respectively. Let $\boldsymbol{B} = (\boldsymbol{M}^{-1})^T\boldsymbol{M}^{-1}$, then

$$\boldsymbol{B} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{f_v^2} & 0 & \dfrac{-\delta_v}{f_v^2} \\ 0 & \dfrac{1}{f_u^2} & \dfrac{-\delta_u}{f_u^2} \\ \dfrac{-\delta_v}{f_v^2} & \dfrac{-\delta_u}{f_u^2} & \dfrac{\delta_v}{{f_v}^2} + \dfrac{\delta_u}{f_u^2} + 1 \end{bmatrix}, \tag{3.25}$$

where $\boldsymbol{B}$ is a symmetric matrix. The two assumptions discussed above can be expressed as

$$(\boldsymbol{h}_1^i)^T\boldsymbol{B}\boldsymbol{h}_2^i = 0. \tag{3.26}$$

and

$$(\boldsymbol{h}_1^i)^T \boldsymbol{B}\boldsymbol{h}_1^i = (\boldsymbol{h}_2^i)^T \boldsymbol{B}\boldsymbol{h}_2^i. \tag{3.27}$$

Similar to the estimation of the initial homographies, the elements of the matrix $\boldsymbol{B}$, can be estimated from the K homographies (see [96] for more details). Given $\boldsymbol{B}$, the elements of the intrinsic matrix can be resolved as in

$$\begin{aligned} f_v &= \sqrt{\frac{1}{B_{11}}} \\ f_u &= \sqrt{\frac{B_{11}}{B_{11}B_{22} - B_{12}^2}} \\ \delta_v &= -B_{13}{f_v}^2 \\ \delta_u &= \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2}. \end{aligned} \tag{3.28}$$

At this stage, the intrinsic matrix, $\boldsymbol{B}$, has been established along side the homography. The extrinsic parameters can be resolved using Eq.3.21 as in,

$$\begin{aligned} \boldsymbol{r}_1^i &= \boldsymbol{M}^{-1}\boldsymbol{h}_1^i \\ \boldsymbol{r}_2^i &= \boldsymbol{M}^{-1}\boldsymbol{h}_2^i \\ \boldsymbol{r}_3^i &= \boldsymbol{r}_1^i \times \boldsymbol{r}_2^i \\ \boldsymbol{t}^i &= \boldsymbol{M}^{-1}\boldsymbol{h}_3^i. \end{aligned} \tag{3.29}$$

Note the computed rotation matrix, $\boldsymbol{R}^i = [\boldsymbol{r}_1^i, \boldsymbol{r}_2^i, \boldsymbol{r}_3^i]$, and translation vector, $\boldsymbol{t}^i$ are still describing and determined by the orientation of the calibration object. In a stereo set up it is useful to establish the extrinsic information in the context of both cameras' location/orientation relative to the other, rather than relative to the calibration object.

Stereo Camera Calibration Stereo calibration builds on the concept of single camera calibration. Determining the relative rotation and translation between the two cameras (in a stereo rig) is simplified given the knowledge of the geometric relationship between each of the cameras and a common calibration object. Consequently, the capturing of the calibration object using the two cameras must be such that both cameras have a large common field of view and the interested visual cues/landmarks (on the calibration object) must be in this common field of view as shown below.

More specifically, to calibrate a stereo rig with a left camera, O_L and right camera, O_R, a dataset $\{(\boldsymbol{x}_1^L, \boldsymbol{x}_1^R, \boldsymbol{w}_1), ..., (\boldsymbol{x}_K^L, \boldsymbol{x}_K^R, \boldsymbol{w}_K)\}$ is required, where $\boldsymbol{x}_k^L$, and $\boldsymbol{x}_k^R$ are the 2D image locations where the 3D point, $\boldsymbol{w}_k$ is observed by O_L and O_R respectively. Subsequently, using the approach discussed above, the set $\{(\boldsymbol{x}_1^L, \boldsymbol{w}_1), ..., (\boldsymbol{x}_K^L, \boldsymbol{w}_K)\}$ is used to establish the intrinsic matrix of the left camera, $\boldsymbol{M}_L$ as well the extrinsic parameters, $\boldsymbol{R}_L$ and $\boldsymbol{t}_L$, that describes the rotation and translation from the camera object coordinate to the left camera coordinate. The set $\{(\boldsymbol{x}_1^R, \boldsymbol{w}_1), ..., (\boldsymbol{x}_K^R, \boldsymbol{w}_K)\}$ can be used to determine $\boldsymbol{M}_R$, $\boldsymbol{R}_R$, and $\boldsymbol{t}_R$ in the case of the right camera. Then the relative rotation, $\boldsymbol{R}$ and translation, $\boldsymbol{t}$ between both cameras can be derived as in

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{R}_R(\boldsymbol{R}_L)^T \\ \boldsymbol{t} &= \boldsymbol{t}_R - \boldsymbol{R}\boldsymbol{t}_L. \end{aligned} \tag{3.30}$$

Description of Registration

Image and depth acquisition were carried out on both the stereo camera and the RGBD sensor, almost adjacently positioned as shown in Figure 3.7. Before capture, the reference stereo camera, O_{M}, and the RGB-Color camera, O_{K}, were stereoscopically calibrated to establish their respective intrinsic and extrinsic parameters . The color channel (as opposed to the depth channel of the RGBD sensor) is used for calibration because one can semi-automatically acquire correspondence matches between it and the reference stereo image pair. Images captured from both cameras were undistorted based on the distortion parameters recovered from calibration. Examining Figure 3.7,

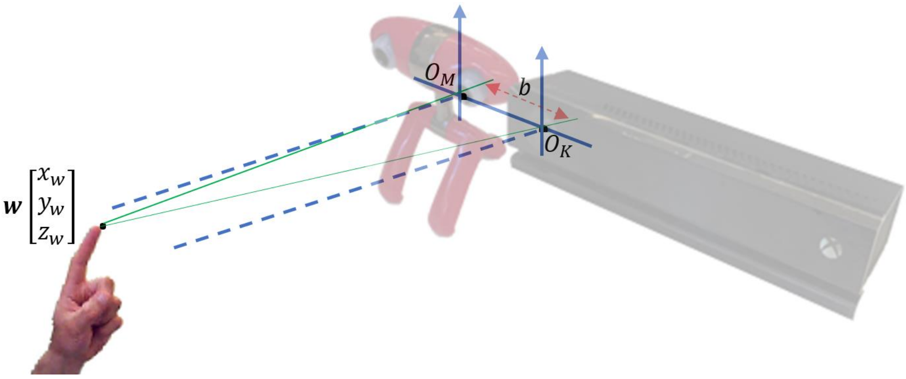


Figure 3.7 Stereo camera and RGBD setup for simultaneous capture of hand pose. The optical center of one of the stereo camera, $\boldsymbol{o}_M$ and that of the RGBD sensor, $\boldsymbol{o}_K$ are positioned adjacently, with both observing points on the hand, $\boldsymbol{w}$. Note how this is set-up yields a separate epipolar geometry and hence knowledge of extrinsic information can be exploited to register both devices.

assume that the observed 3D point $\boldsymbol{w}$, was projected onto the reference [1] stereo camera and the RGBD-Color image plane at the points, $\boldsymbol{x}_M = [u_M, v_M]^{\mathrm{T}}$ and $\boldsymbol{x}_K = [u_K, v_K]^{\mathrm{T}}$. First, the points in the RGBD-Color plane are back-projected into 3D by applying its previously calibrated projection matrix and the accompanying depth information (courtesy of its depth channel) as in,

$$\boldsymbol{p}_K = \boldsymbol{M}_K{}^{-1} \begin{bmatrix} u_K \\ v_K \\ 1 \end{bmatrix} z_w, \tag{3.31}$$

where $\boldsymbol{M}_K$ is the intrinsic matrix of the RGB-Color camera and $\boldsymbol{p}_K$ is the coordinate of the 3D point, $\boldsymbol{w}$ relative to O_{K}. The 3D projection, $\boldsymbol{p}_K$ is then transformed into the reference stereo camera coordinate using the relative rotation, $\boldsymbol{R}_{\mathrm{KM}}$ and translation,

[1] The reference stereo image is one of the two images in the pair. For each pixel in the reference image, we seek a correspondence in the other image. Hence a resulting disparity image register perfectly with the reference stereo image.

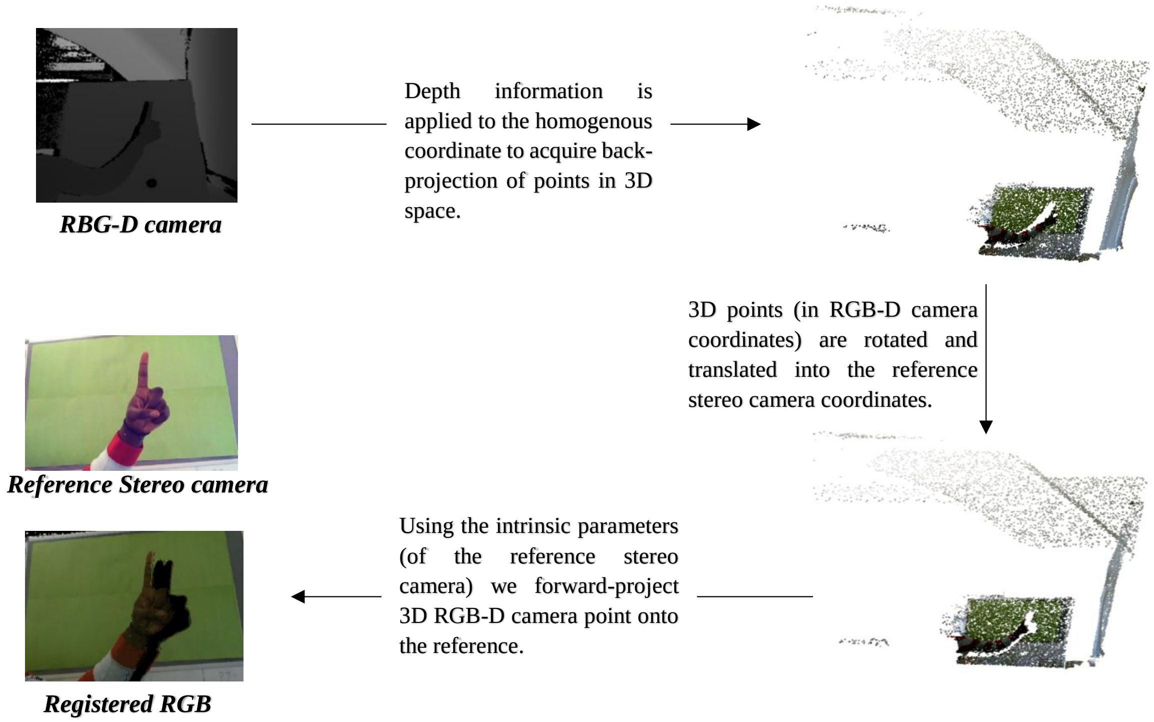

RBG-D camera
Depth information is applied to the homogenous coordinate to acquire back-projection of points in 3D space.
3D points (in RGB-D camera coordinates) are rotated and translated into the reference stereo camera coordinates.
Reference Stereo camera
Using the intrinsic parameters (of the reference stereo camera) we forward-project 3D RGB-D camera point onto the reference.
Registered RGB

Figure 3.8 Transferring the depth data from an RGBD camera to establish groundtruth depth for the stereo data. Hand poses are captured simultaneously using adjacently positioned calibrated stereo camera and RGBD camera. First, all 2D positions on the RGBD camera are back-projected to 3D. By applying the rotation and translation matrix between the RGBD camera and the reference stereo camera we transform points in the RGBD camera to the camera coordinates of the reference stereo camera. Lastly, we forward project these points onto the reference stereo camera image plane by using its projection matrix. In effect, depth values of the RGBD image are transferred to the reference stereo image, forming groundtruth for training the unary term.

$\boldsymbol{t}_{\mathrm{KM}}$ information between O_{K} and O_{M},

$$\begin{bmatrix} \boldsymbol{p}_M \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{KM} & \boldsymbol{t}_{KM} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_K \\ 1 \end{bmatrix}. \tag{3.32}$$

Finally, $\boldsymbol{p}_M$ is forward-projected into the reference stereo camera plane as in,

$$\lambda \begin{bmatrix} u_M \\ v_M \\ 1 \end{bmatrix} = \boldsymbol{M}_M \boldsymbol{p}_M, \tag{3.33}$$

where $\boldsymbol{M}_M$ is the intrinsic matrix of the reference stereo camera and λ is the third component of the resulting dot product, $\boldsymbol{M}_M \boldsymbol{p}_M$. The consequence of Eq. 3.31, Eq. 3.32 and Eq. 3.33 is that a mapping from the RGB-Color (with accompanying depth information) to the reference stereo camera is established. Thus, the depth information from the depth channel of the RGBD sensor is well registered with the reference stereo image and hence can be later used as groundtruth for learning and evaluating depth recovery solely from stereo captures (see Figure 3.8).

3.2 Machine Learning

The goal of this thesis is to estimate hand pose (and depth, as an intermediate representation) from stereoscopic images of hands. In the previous section, the key concepts for lower level data capture were introduced. This section focuses on higher level inference that will be key to depth and pose estimation later in this thesis.

3.2.1 Probabilistic Modeling

Probability is the mathematical language of describing the propensity and uncertainty of an event. A *random variable* is a variable with an uncertain quantity. This could either be *continuous*, where it could be of any real number value, or *discrete*, where there is a pre-defined number of values it could take. Often it is important to examine all the possible values that a random variable could take and explore the relative probability of each of these states, for instance, all the possible poses that a hand could take. This information is presented in a probability distribution, or a *probability density function* and a set of all the possible values of the random variable is referred to as the *probability space.* In some scenarios, it is useful to examine the probability of multiple random variables in tandem. In this case, a *joint probability* can be established where each point in the joint probability space is the likelihood of each variable simultaneously having specific values. Lastly, the probability of one random variable could be of interest given the knowledge of another, in this case, a *conditional probability* is defined. Consider a random variable, x, with a probability, $Pr(x)$, and a second random variable, y with a probability, $Pr(y)$, the joint probability of both variables x and y is represented as $Pr(x, y)$ whilst the conditional probability of the variable x given knowledge of y is represented as $Pr(x|y)$. For continuous variables, $\int_y \int_x Pr(x, y)dxdy = 1$ and $\int_x Pr(x|y)dx = 1$ whilst in the discrete case $\sum_i \sum_j Pr(x_j, y_i) = 1$ and $\sum_i Pr(x_i|y)dx = 1$.

Marginalization and Factorization

Given the joint distribution, $Pr(x, y)$, it is often useful to determine the probability of the random variable, $Pr(y)$, as in

$$Pr(y) = \int_x Pr(x, y)dx. \tag{3.34}$$

This is referred to has *marginalization* of the joint probability, $Pr(x, y)$. A more common situation in this thesis, is the calculation of the joint distribution between an observed quantity, x (e.g. a pixel intensity value) and a quantity to be determined, y (e.g. the hand pose). This requires establishing the conditional probability, $Pr(y|x = x_o)$, where x_o is the observed value of x. This is referred to as the *factorization* of the joint probability, $Pr(x, y)$.

Expectation

Expectation, or the expected value of a variable is the frequency based weighted mean of all the possible values the variable could have. More precisely, the expectation of a variable, X, is defined as $\mathbb{E}(x) = \int_{-\infty}^{\infty} xPr(x)dx$, in a continuous case and $\mathbb{E}(x) =$

$\sum_{-\infty}^{\infty} x_i Pr(x = x_i)$ in a discrete case. Expectation is significant because probabilities can be represented as expectations. For instance $Pr(x \in X) = \mathbb{E}[\mathbb{I}(x \in X)]$, where $\mathbb{I}[]$ is an indicator function that returns 1 if the argument is true and returns 0 otherwise.

3.2.2 Learning and Inference

Learning in the context of machines is a concept largely transferred from humans. Like in humans, learning involves establishing higher level connections between different concepts and entities, with the aim of applying this in other previously unseen scenarios. This requires an extrapolation of knowledge from example scenarios (of which one has high certainty) to less certain scenarios/concepts. For instance, a baby extrapolating for a general concept of cars from a small labelled set of images of cars. Hence, the degree of certainty (or uncertainty) and subsequently probability, plays a significant role in learning.

More formally, consider the inference task of establishing the mapping from an input space to a target space, $f : \mathcal{X} \mapsto \mathcal{Y}$, using a dataset of K examples, $\mathcal{U} = \{(x_1, y_1), ...(x_K, y_K)\}$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The set, $\mathcal{U}$, is referred to as a groundtruth because there is an absolute degree of confidence that the pairings in the set are indeed correct. Now the established f can be used to predict for the target, y^*, of a previously unseen data point, $x' \in \mathcal{X}$, with some degree of certainty.

Classification: In a scenario whereby, the variable in the target space is discrete, the learning problem is referred to as *classification.* This is often not the case in this thesis.

Regression: If the target variable can take a continuous range of values then the learning is referred to as *regression.* This is largely the case in this thesis. For instance the inference of the continuous 3D position of hand joints given an input depth image.

As previously introduced, probability/uncertainty are very significant in a learning framework. Hence it is useful to fit probability models to a dataset as a learning process. Specifically, the parameters of the probability model, $\boldsymbol{\theta}$, are learned such that the probability model can be sampled to recreate the training dataset. Two main approaches to this are presented in the following discussion, namely *maximum likelihood*, ML, and *maximum a posteriori*, MAP [57].

Maximum Likelihood

Given a training dataset, the task is to establish the model parameters, $\hat{\boldsymbol{\theta}}$, such that the joint probability of all data points in the dataset based on the model is maximized. Continuing from the previously described learning task above, the likelihood function can be represented as $Pr(y_k|x_k, \boldsymbol{\theta})$. The maximum likelihood estimation of model parameters can be formulated as

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}}[\prod_{k=1}^{K} Pr(y_k|x_k, \boldsymbol{\theta})], \tag{3.35}$$

where $\boldsymbol{\theta}$ denotes the parameters of a generic probability model.

Maximum a Posteriori

In contrast to maximum likelihood, maximum a posteriori models the probability of the model parameters given the dataset, $Pr(\boldsymbol{\theta}|\{y_k, x_k\})$, by introducing a prior distribution over the model parameter, $Pr(\boldsymbol{\theta})$. Hence the Maximum a Posterior estimation of the model parameter can be derived as follow:

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} \left[\prod_{k=1}^{K} Pr(\boldsymbol{\theta}|\{y_k, x_k\})\right], \\ &= \arg\max_{\boldsymbol{\theta}} \left[\frac{\prod_{k=1}^{K} Pr(\{y_k, x_k\}|\boldsymbol{\theta})Pr(\boldsymbol{\theta})}{Pr(y_k, x_k)}\right],\end{aligned} \tag{3.36}$$

where Bayes' rule is applied between the first and the second line. For the task of maximization, it suffices to ignore the denominator as in

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left[\prod_{k=1}^{K} Pr(\{y_k, x_k\}|\boldsymbol{\theta})Pr(\boldsymbol{\theta})\right]. \tag{3.37}$$

3.2.3 Stochastic vs Deterministic Models

In this thesis, the first step in the machine learning approach taken to address different stereo hand pose based problems is to determine a mathematical model in which the relationship between the real-world quantities are represented. These models are often classified as either *deterministic*, *stochastic*, or a *hybrid* of both. A model is considered to be deterministic if the state of all its properties can be conclusively determined. A model is considered to be stochastic if it contains elements of randomness. The consequence of this is that under the same set of inputs, a stochastic model will yield different predictions at different instances of being run. The opposite is the case when deterministic models make predictions. A benefit of a stochastic model is that an analytical solution is not required. Hence, it is useful to represent systems that are too cumbersome to solve for analytically in a stochastic model.

3.2.4 Random Forest

Random forest is a member of a wide range of voting based ensemble models. Specifically, it is an ensemble of randomly trained decision trees. Each decision tree establishes a non-linear mapping between high dimensional input and target spaces. Before delving deeper into the intricacies of random forest, it is useful to explore a significant building block - decision trees.

Decision Trees and Forest

Decision trees are an explicit and more visual modeling of decision-making process during learning. A decision tree is a collection of condition statements (often referred to as *nodes*)that are relatively related in a cascading manner - forming a tree-like architecture. Each node will often posses multiple children nodes and so on and the task of "making a decision" over an unknown entity with consists of evaluating the entity based on these condition statements starting from the uppermost node. The result of this evaluation will determine which of the children nodes is evaluated for next.

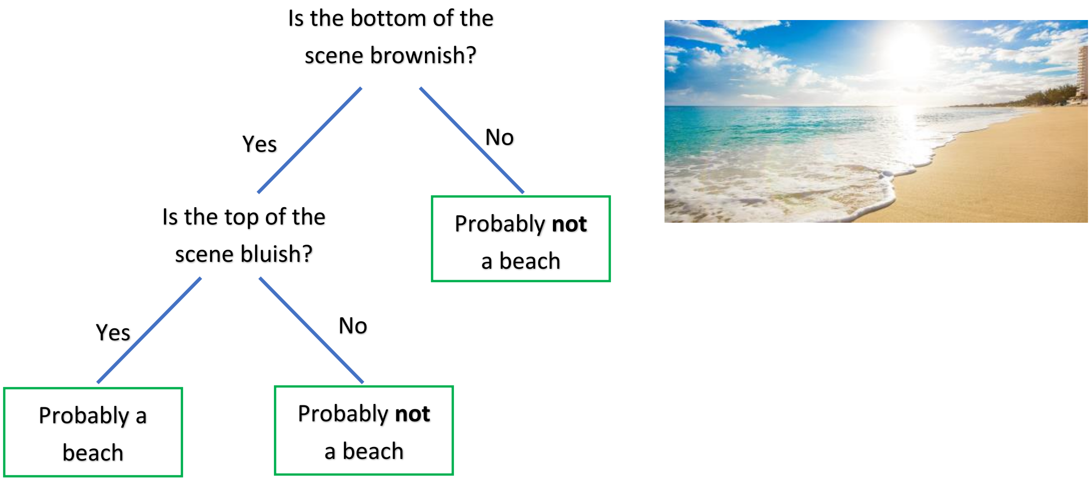


Figure 3.9 An hypothetical decision tree for detecting images of beaches. A concatenation of numerous simple features is used arrive at a strong classifier. The green boxes indicate terminating leaf nodes whilst the blue lines indicates the branches of each node on the tree.

Hence a decision tree can be considered as a combination of several weak descriptors of the entities to be evaluated. Whilst these features are weak (in that they might not on their own help describe or discriminate the type of entity), a combination of them can be applied in a decision tree set-up to achieve strong evaluation. Consider the task of making a decision on whether or not a presented image is that of a beach, Figure 3.9 illustrates a hypothetical decision tree to determine this. The first node tests the color of the base of the image for a light brownish color, depending on the result of this condition, one of its children not is evaluated subsequently on this the end of the tree is reached. The terminating node of the tree is where the final decision on the entity is made and it is often referred to as the *leaf node* (in Figure 3.9 these are indicated with the green boxes).

To improve the robustness of the model the prediction (decision) from a collection of unique decision trees can be combined. This model is referred to as a *decision forest*. Here each tree in the forest is exposed to a subspace of the dataset domain. In the case of a random forest, its decision trees are exposed to randomly selected non-intersecting subsets of the training dataset, the decision trees are trained independently. Then at test time, the prediction is based on the aggregation of the posterior distribution of each tree. The effect of this is a highly generalized model with robustness to over-fitting.

Regressive Random Forest

As with most machine learning frameworks, random forest handles regression as well as classification based problems. In this thesis, Random Forest is used predominantly for regression. This is typified by the work presented in Chapter 4. The task is to improve the disparity map recovered from stereo capture by establishing a mapping from this disparity to robust groundtruth depth values.

Consider a learning task similar to that introduced in Section 3.2.2, except that the input space is a multidimensional space such that there is a vector based input variable, $\boldsymbol{x}$, and consequently a dataset, $\mathcal{U} = \{(\boldsymbol{x}_1, y_1), ...(\boldsymbol{x}_K, y_K)\}$. The task is to model $\boldsymbol{x}_k \mapsto y_k$ using a Random Forest consisting of T trees. First, a subset of the dataset, $\mathcal{U}_t \subset \mathcal{U}$ is established by random sampling and assigned to each tree, t such that $\{\mathcal{U}_t\}_{t=1}^{T}$ are generated with replacement hence they might have overlapping members. A decision tree consists of a hierarchy of split nodes and each tree is grown in training by recursively splitting training data (fed into each node) into two disjointed subsets that are passed onto two subsequent sub-nodes. The splitting criteria is based on the value of a randomly selected index of the input vector, $\boldsymbol{x}$, and a threshold value. Optimal splitting parameters, $\theta = \{w, \tau\}$ are stored for each node of the tree, where w is the index of the member of $\boldsymbol{x}$ to be evaluated and τ is the threshold value used to split data. More formally, for an inbound set of data, S_i in the i^{th} node, this is evaluated on a splitting function, $F(S_i, \theta_i)$ as in

$$F(S_i, \theta_i) = \begin{cases} S_{i,R} = \{\boldsymbol{x}\} : \boldsymbol{x}[w] > \tau_i \\ S_{i,L} = \{\boldsymbol{x}\} : \boldsymbol{x}[w] <= \tau_i \end{cases}, \tag{3.38}$$

where $S_{i,L}$ and $S_{i,R}$ indicate the disjointed subsets of S_i that is passed on the left and right subsequent sub-node respectively. The optimal splitting criteria is determined based on the information gain, $Q\{F(S_i, \theta_i)\}$, defined as

$$Q\{F(S_i, \theta_i)\} = H\{S_i\} - \sum_{b \in \{L,R\}} \frac{|S_{i,b}|}{|S_i|} H\{S_{i,b}\}, \tag{3.39}$$

where $H\{\}$ represents the entropy of a set. In a regression forest a differential entropy is used as in

$$H\{S\} = -\frac{1}{|S|} \sum_{\boldsymbol{x} \in S} [\int_y Pr(y|\boldsymbol{x}) log\{Pr(y|\boldsymbol{x})\} dy]. \tag{3.40}$$

To simplify derivation, a Gaussian distribution could be assumed, in which case the the posterior probability, $Pr(y|\boldsymbol{x})$ can be represented as

$$Pr(y|\boldsymbol{x}) \sim \mathscr{N}(\bar{y}, \sigma_s), \tag{3.41}$$

where $\bar{y}$ and σ_s are the mean and the variance of the target elements of the set S [97]. Hence Eq. 4.3 can be rewritten as

$$H\{S\} = -\frac{1}{|S|} \sum_{\boldsymbol{x} \in S} log\{\sigma_s\}. \tag{3.42}$$

The recursive splitting is continued until a level of entropy is reached or other criteria are met. At this stage, the terminating nodes are referred to as *leaf* nodes. A statistical analysis is done on the target elements of data points that reach each leaf node. A typical analysis used in this thesis is the probability distribution of leaf node target data points. At test time, where the task is to determine the target value of a previously unseen input vector, $\boldsymbol{x}'$, the data point is propagated through all trees in the forest evaluated by the splitting function at each node until a leaf node is reached. Each tree, t, gives a posterior probability, $Pr_t(y|\boldsymbol{x}')$, which can be aggregated across all trees as in

$$Pr(y|\boldsymbol{x}') = \frac{1}{T} \sum_{t=1}^{T} Pr_t(y|\boldsymbol{x}'). \tag{3.43}$$

The target prediction of $\boldsymbol{x}'$ can be determined by Maximum Likelihood Estimate (MLE) as in

$$y^* = \arg\max_y Pr(y|\boldsymbol{x}'). \tag{3.44}$$

3.2.5 Convolutional Neural Network

Unlike Random Forests, convolutional neural networks require no hand-crafted features, that are dependent on problem-specific expertise to design a feature extractor that transforms raw data into representative feature vectors. These features are used in learning an algorithm to infer or classify patterns within a given problem domain. The performance of these features is limited to human knowledge about the problem. CNNs are able to learn features that are oblivious to human intuition. This is a desirable trait of a CNN, however, this creates a non-linear space with high learning complexity. This makes it inherently difficult to generalize, particularly with limited training data. CNNs are extensions of Artificial Neural Networks (ANN)s that consist of layers of neurons described by their weights and biases. CNNs unlike ANNs are such that some neurons in a prior layer are not connected to neurons in the next layers. This yields a convolving mask implementation. Figure 3.10a shows a typical structure in an ANN in comparison to that of a CNN, presented in Figure 3.12. The ANN receives data (from the input space) as a vector in its input layer neurons. This data is multiplied with unique weight values and fed into an *activation function* before being passed onto every neuron in the next layer (the hidden layer). This data is propagated into the second hidden layer and then the output layer in a similar manner. Note that there is a weighting between all possible pairs of input layer neuron and hidden layer neurons. The values that are propagated out of the output layer are the predictions of the ANN given the input vector. Consider the case where the input space is an image. For instance, a $480 \times 640 \times 3$ image will yield $921,600 * K$ weights to connect to an immediate hidden layer with K neurons. This is a large number of weights to learn. A rectification to this is to drop some of the weights as in Figure 3.10b. Consequently, not all the neurons in a layer are connected to the neurons in the succeeding layer.

In conjunction with dropping the weight, weight sharing could be applied to further reduce the number of weights. Hence rather than having unique weights between each pair of neurons, the same weight value could be applied to another pair of neurons. Now examine the setup in Figure 3.11a, where the same structure in Figure 3.10b is maintained, however, each distinct weight has been allocated a distinct color. The effect of weight sharing is illustrated in Figure 3.11b. For instance, note how in between the input layer and hidden layer 1, only the first two weights (indicated in green and blue color) are used, after weight sharing. Similarly between hidden layer 1 and hidden layer 2 three weights are used, illustrated in purple, black and brown color. This manifests into convolving an array of neurons (or pixels in the case of images) with a kernel. The consequence of this is a convolutional network configuration as opposed to a fully connected configuration presented in Figure 3.10a. A more complete illustration of a CNN is presented in Figure 3.12.

CNNs like ANNs, aim at establishing a non-linear mapping from an input space to a target space. In the case of CNNs, this input space tends to be images. For instance, the CNN presented in the figure above illustrates a CNN that aims to establish the relationship between an image of a hand pose and a vector of elements that describe

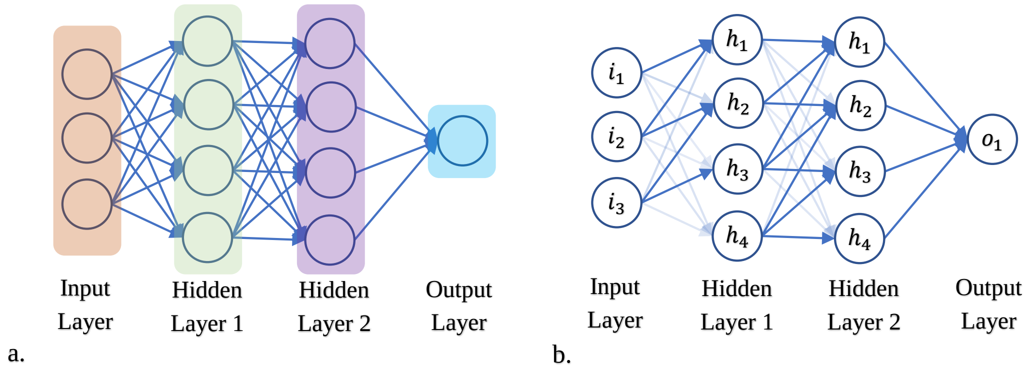


Figure 3.10 (a). Fully connected Artificial Neural Network (ANN) Structure, where each arrow indicates a weight connecting a neuron in a prior layer to another in the succeeding layer. (b). Illustration of weight dropping on a fully connected ANN. The faintly coloured arrow indicate dropped weights.

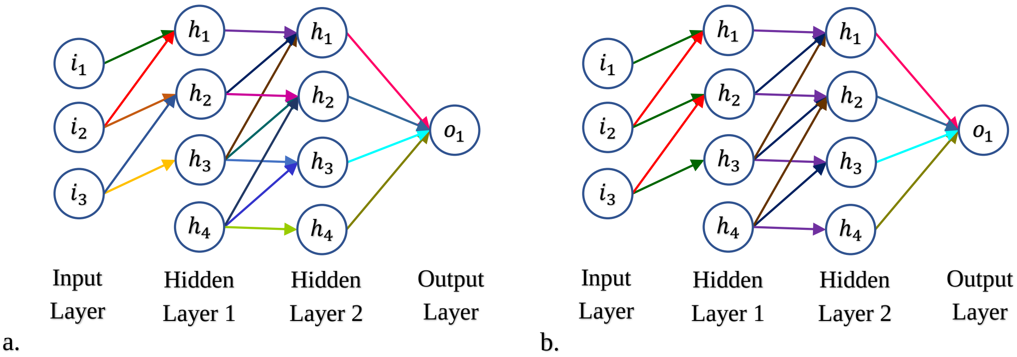


Figure 3.11 (a) Illustration of an ANN after weight dropping. Observed how the dropped weights yields a less complex network. (Note each distinct colored arrow indicates distinct weights) (b) Illustration of an ANN after weight dropping and sharing. Observe the color of the arrows and note how same weights are shared along different neuron pairs, yielding the effect of sliding (convolving) a mask along an array of data points.

the pose of the hand. In this case, $[j_{i,x}, j_{i,y}, j_{i,z}]^T$ represent the 3D spatial position of the i^{th} joint of the hand i.e. distal little (pinky) finger joint, intermediate ring finger joint etc. Unlike ANNs, CNNs consist of different unique types of layers which in most cases are Convolution, Pooling, and Fully-connected, as illustrated above. The convolution layer computes the outputs that are fed into its succeeding layer by taking a dot product of the shared weights and a sub-region of its input. The Pooling layer will spatially down-sample the input to yield a smaller sized output. Finally, the fully

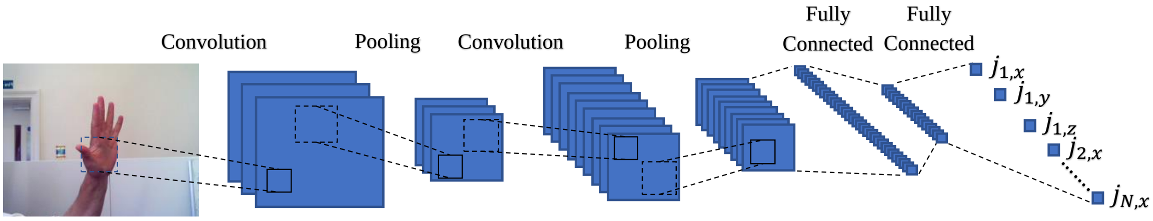


Figure 3.12 A Typical structure of a Convolutional Neural Network. Here the input image is first convolved with pre-defined masks before undergoing a pooling (down-sampling) operation. This constitutes a convolution layer. After passing the inputting image through multiple convolution layers, the resulting image is concatenated into a vector and passed into fully-connected neuron layers to result in a prediction vector of the desired length.

connected layer operates as a conventional ANN as introduced above. As previously stated the output of each layer is fed into an activation element-wise before being propagated onto the following layer. In the case of a CNN, this is often either a sigmoid function, $\sigma(x)$, or a Rectified Linear Unit (ReLU) function, $g(x)$, where

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{3.45}$$

and

$$g(x) = \begin{cases} x & x > 0 \\ 0 & x <= 0 \end{cases}. \tag{3.46}$$

Both functions introduce a non-linear transformation to the mapping from the input space to the output space.

Back Propagation

With an established network structure, the next task is to establish the values of the weights such that the mapping between the input and output space is consistent with a groundtruth dataset. Given a training data pair set, $\{\boldsymbol{I}_1, \boldsymbol{j}_1\}, \{\boldsymbol{I}_2, \boldsymbol{j}_2\}...\{\boldsymbol{I}_K, \boldsymbol{j}_K\}$ with an input image, $\boldsymbol{I}_k$, and a target vector, $\boldsymbol{j}_k$, the goal is to establish the set of weights, $\mathcal{W} = \{W_1^{1,2}, W_2^{1,2}, ..., W_1^{2,3}, W_2^{2,3}, ..., ...\}$ such that

$$\arg\max_{\mathcal{W}} \sum_{k=1}^{K} \phi\{F(\mathcal{W}, \boldsymbol{I}_k), \boldsymbol{j}_k\}, \tag{3.47}$$

where $\phi\{\}$ is a loss function between the CNN predicted output (from propagating data from the input image through the network to the output layer) and the groundtruth output, $\boldsymbol{j}_k$. $F()$ describes the structure of the architecture of the network and how the weights interact with the input image. To solve for Eq. 3.47 the error value outputted by $\phi\{\}$ is propagated back into the network to update $\mathcal{W}$ this error is reduced. This is repeated until a desirable error is achieved. This is referred to as *back propagation*. The key to back propagation is to establish the partial derivative of the error with respect to each weight, $\frac{\delta E}{\delta W_\alpha} : \forall W_\alpha \in \mathcal{W}$. In this thesis, the weights of the CNN are updated using *gradient descent* as in

$$W_\alpha \leftarrow W_\alpha - \eta \frac{\delta E}{\delta W_\alpha} - \eta \lambda W_\alpha, \tag{3.48}$$

where η is the *learning rate* and λ is the *weight decay.* Note that in the work implemented in this thesis, the MatConvNet framework was used [2], which comes with gradient descent implementation.

3.2.6 Conditional Random Field

The input-target paradigm of learning and inference is very successful, however, it assumes that each of the data pairs is independent of each other. In some cases, this assumption does not hold true. Knowledge of the state of some variables will often improve the confidence in predicting the state of others. The image-based nature of the problem being addressed in this thesis inherently lends itself to the scenario where states of neighboring pixels are often similar. To this end, it is useful to model the probability of the state of a pixel conditioned on the neighboring pixels. A convenient way of modeling this is via *graphical modeling*, described below.

A significant consequence of graphical modeling is that, in effect, the probability distribution over the variables of a multi-variable system (with many complex inter-variable interactions) can be derived from the product of *factors* of a much smaller subset of the variables, where the factors indicate the probability of a subset of neighbouring variables taking a particular joint assignment. The joint probability of the set of variables in the entire network is hence the product of the joint factors of all the sub-groupings in the graph. To maintain a probability distribution (which sums to 1), the probability is divided by the partitioning function, Z. This yields a normalized probability distribution.

Consider a system possessing a set of observable random variables, $X = \{X_1, ..., X_{|X|}\}$, such as pixel intensity values and a set of target variables, $Y = \{Y_1, ..., Y_{|Y|}\}$, like the

[2] http://www.vlfeat.org/matconvnet/

depth at each pixel in an image of interest. The task is to establish a joint probability distribution over all members of set Y given a knowledge of all members of set X, assuming a set of factors, $\mathcal{W} = \{\psi_1, ..., \psi_N\}$, is established and that the vectors $\boldsymbol{x} = [x_1, ..., x_{|X|}]^T$ and $\boldsymbol{y} = [y_1, ..., y_{|Y|}]^T$ are vectors of actual allocations values to the variables in the sets X and Y respectively. The conditional distribution, $Pr(\boldsymbol{y}|\boldsymbol{x})$, can then be represented as

$$Pr(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{n=1}^{N} \psi_n(X_n, Y_n), \tag{3.49}$$

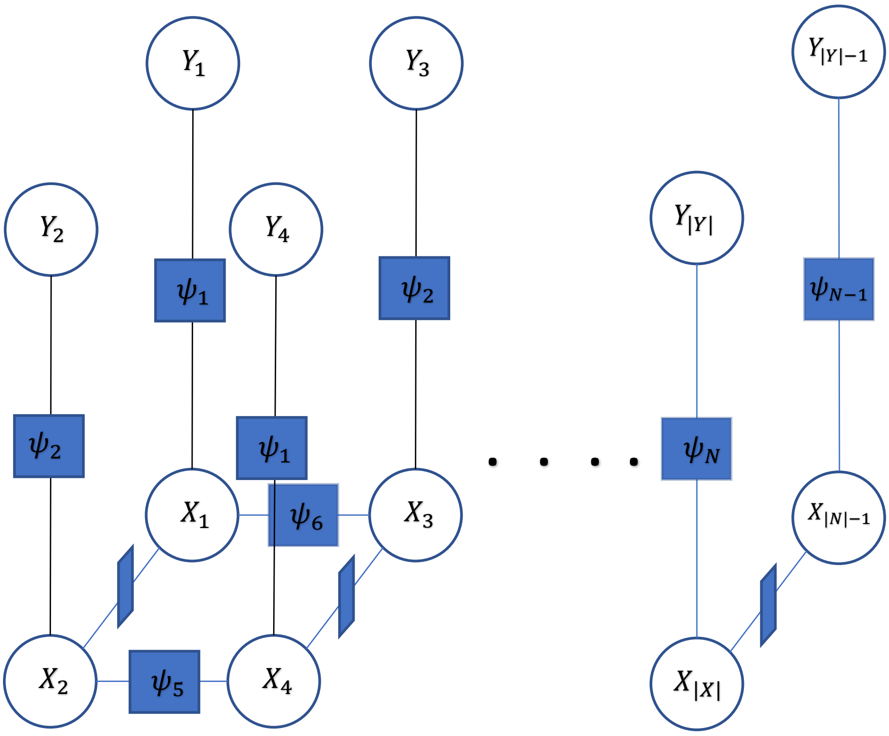


Figure 3.13 Hypothetical Factor Graph. The circles represents variables whilst the squares represents factors. The X label represents observable variables (e.g. intensity value of a pixel) and Y is used for variables whose state is to determined (e.g. depth value at a pixel). Each factor represents the probability of the set of connecting variable having different joint assignment.

where $X_n \subseteq X$ and $Y_n \subseteq Y$, are unique subsets of the observable, N is the number of factors, and target objects whose probability of joint assignment is represented by the factor ψ_n. The partitioning function, $Z(\boldsymbol{x})$, is the normalising term:

$$Z(\boldsymbol{x}) = \int_{\boldsymbol{y}} \prod_{n=1}^{N} \psi_n(X_n, Y_n), \tag{3.50}$$

Figure 3.13 gives a graphical illustration of the system introduced, whereby the graph, $G = (V, F, E)$, the vertex set, $V = X \cup Y$, the set of factors, $F = \mathcal{W}$, and the set of edges, E are shown, defining the scope of each factors. For instance the scope of factor ψ_5 is X_2, X_4. Note how each factor is a local function whose scope is the subset of variables, $\{X_n \cup Y_n\}$ whose joint assignment probability it describes. However, an aggregation of these local compatibility functions is derived in Eq. 3.49 to provide a joint probability distribution for all members of set Y given a knowledge of all members of set X. The work presented in Chapter 6, relies heavily on the concept of a CRF. Here the task of superpixel based robust depth estimation from disparity information is addressed with a Random Forest modeling the mapping between per-superpixel disparity information to depth in a naive manner. The predicted posterior probability is augmented with a CRF model that merges it with pairwise neighboring superpixel factors.

3.2.7 Markov-chain Monte Carlo

To review the concept of Markov-chain Monte Carlo, it is particularly useful to discuss the two fundamental concepts it is based upon, namely: *Monte Carlo approximation* and *Markov Chains.*

Monte Carlo Approximation

Monte Carlo is a method of approximating the state of a model that is too complex to compute deterministically and with an intractable solution. Assume a model whose state is described by the probability distribution function, $f(X)$. The goal here is to

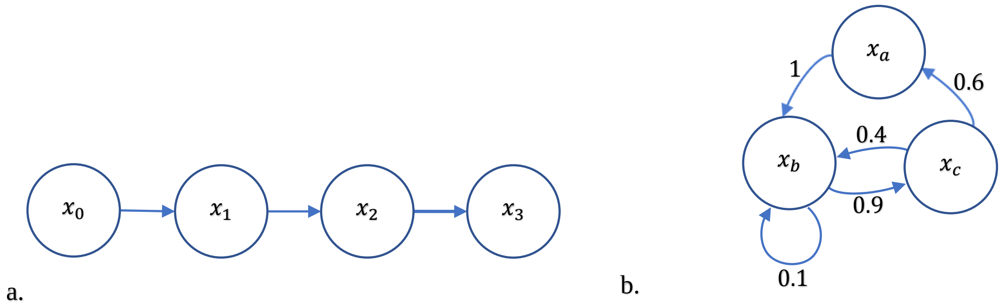


Figure 3.14 (a) Sequence of events (b) A discrete example of a non-trivial Markov chain.

approximate the expectation of the state of the model (distribution function), $\mathbb{E}[f(X)]$. The Monte Carlo approximation approach is based on the *law of large numbers*, which states that as the number of observations (samples) of the state of a system increases, the average becomes closer to the expectation. Hence, the Monte Carlo estimator to the expectation problem above is defined as

$$\mathbb{E}[f(X)]) = \int_{-\infty}^{\infty} X f(X) dX = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} f(X_i), \tag{3.51}$$

where $X_1, ..., X_N$ are randomly sampled from an arbitrary distribution, $g(X)$. It is necessary for N to be substantially large to approximate the expectation. The consequence of this is that samples based on a much simpler distribution (e.g. a uniform distribution) can be evaluated under $f(X)$ and used to estimate $\mathbb{E}[f(X)]$ by computing an average of the evaluation output.

Markov Chains and Ergodic Theorem

A Markov Chain is a sequence of events that satisfies the Markov property that each event in the sequence is solely dependent on the preceding event. Consider the following sequence of events in Figure 3.14a. It is considered a Markov chain if

$$Pr(x_3 | x_2, x_1, x_0) = Pr(x_3 | x_2). \tag{3.52}$$

Consequently, knowledge of the prior state of a sequence of events is the most informative one can get to predict the succeeding event. Figure 3.14(b) illustrates a less trivial Markov chain. Here a transition can occur between all three possible states and the likelihood of each transition is presented as probabilities assigned to each arrow. For instance the probability of system transitioning from state x_b to x_c is 0.9; from x_a to x_b is 1 etc. This information can be represented in a *transition matrix* as in

$$\boldsymbol{T} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}. \tag{3.53}$$

If $\boldsymbol{\phi}_t = (x_{a,t}, x_{b,t}, x_{c,t})$ is the probability distribution of the system at time, t, where $\boldsymbol{\phi}_0$ is the prior probability distribution of the system, then $\boldsymbol{\phi}_{t+1} = \boldsymbol{T}\boldsymbol{\phi}_t$. After several iterations, the probability distribution converges to $(0.2, 0.4, 0.4)$, regardless of the prior distribution, ϕ_0. This distribution is referred to as the *stationary distribution* of the Markov chain and it is the overall probability of being in each of the states x_a, x_b, and x_c over an infinite number of transitions. Based on this attribute, the Markov chain presented in Figure 3.14 is said to be *ergodic*. Consequently, an ergodic Markov chain modeled system is such that, regardless of the starting state of the system, every possible state will eventually be transitioned to in a finite time, and more importantly, the number of times each state is transitioned to is proportional to the probability of such state. This is a significant property in the context of Markov Chain Monte Carlo (MCMC) as will be discussed. Two conditions for a Markov chain to be ergodic are that the Markov chain graph must of *irreducible* and *aperiodic*. Irreducible meaning there is a non-zero probability of transitioning from any possible state to another, whilst aperiodic implies that "there is no integer $k > 1$ that divides the length of every cycle in the graph" [98].

Markov-chain Monte Carlo

Markov-chain Monte Carlo (MCMC) is a tool for sampling from (and also computing expectation of) very complex and high dimensional probability distributions. MCMC is largely based on the Monte Carlo sampling intuition however instead of using randomly sampled elements, it uses a sequence $X_1, ..., X_N$ drawn from a Markov chain. Consider the task of sampling from a complex probability distribution, π. The first step will be to construct a Markov Chain with a stationary distribution, π, and then the ergodic theorem is applied. More specifically, it sequentially samples points by conditioning on the current location and Markov chain transition matrix. Given that the Markov chain remains ergodic, the distribution of the sequence of samples converges to π. Consequently, MCMC exploits the fact that in a probability distribution space, regions of high probability are also neighboured by regions of high probability distributions and the regions of low probability are neighboured by regions of low probability distributions.

3.3 Datasets

The frameworks presented in the latter part of this thesis were trained and validated on three datasets. This section introduces these datasets and how they were collected. The first two datasets (Dataset A and Dataset B) address the problem of hand depth from stereo capture whilst the third dataset (Dataset C) addresses pose estimation from stereo capture.

3.3.1 Dataset A

The first dataset was relatively simple consisting of data pairs of stereo and depth. Five participants were involved, a total of 1,000 (200 per participant) captures were made, each articulating seven poses. Variation in poses consisted of simple finger flexion and extension with the relative hand to camera orientation kept the same at the fronto-parallel view.

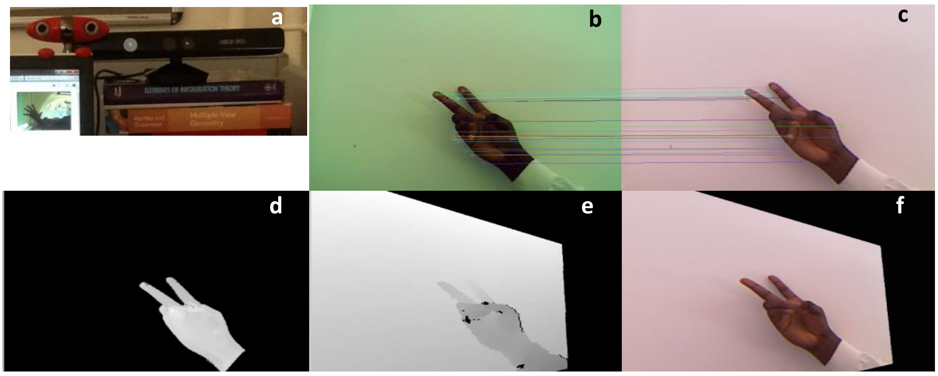


Figure 3.15 The stereo camera and the RGBD sensor are positioned adjacently (a). First correspondence between both cameras is acquired (b, c), which is used to estimate the projective transformation that achieves registration (f). This transformation is then applied to the depth image (e). The process results in a disparity image (d) acquired from stereo-matching, and a closely registered corresponding groundtruth depth (e).

The key challenge in the dataset acquisition was mapping the disparity image data to groundtruth depth data, so as to establish a strong correspondence between the pairs of data. An alternative method to that presented in Section 3.1.3 was required as the checkerboard pattern data was not available for this dataset. To achieve this, image and depth acquisition were carried out on both the stereo camera and the active depth sensor as shown in Figure 3.15. As discussed in Section 3.1.3, the RGBD sensor used acquires RGB data that is well aligned with the depth channel. First, Scale Invariant Feature Transform [99] features were extracted from the reference stereo image and the RGBD depth channel. This, in turn, was used to establish corresponding points

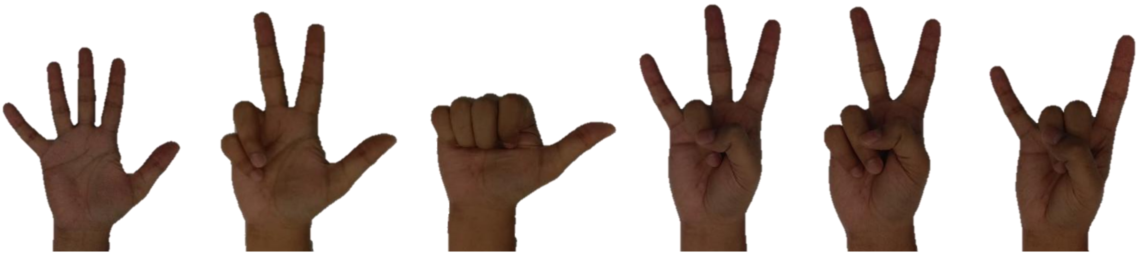


Figure 3.16 The six different poses captured in Dataset A. Note how these posses are solely in front-parallel view.

and hence the geometric transformation that registers the RGBD depth channel to the reference stereo image. The (RANSAC) [95] algorithm was used to establish the optimum geometric transformation. The mean residue error of all inlier points was computed as $\frac{1}{I}\sum_i (\boldsymbol{p}_i^R - \hat{\boldsymbol{G}}\boldsymbol{p}_i^L)^2$, where $\boldsymbol{p}_i^R$, and $\boldsymbol{p}_i^L$ are corresponding points in both images planes; $\hat{\boldsymbol{G}}$ is the resolved geometric transformation; and I is the total number of inlier points. Under this condition, the mean residue error of Dataset A was 5.425 pixels. The established transformation is then applied to the acquired depth image, resulting in an alignment and registration of the reference stereo image to the depth image from the RGBD sensor.

3.3.2 Dataset B

The second dataset consists of more complex poses where the relative hand camera orientation is not just fronto-parallel but arbitrarily orientated. Consequently, a geometric mapping will not suffice to establish RGBD-stereo registration. Instead the approach introduced in Section 3.1.3 was used. The stereo-RGBD calibration was done with a re-projection error of $9.864mm$. This re-projection error is based on the

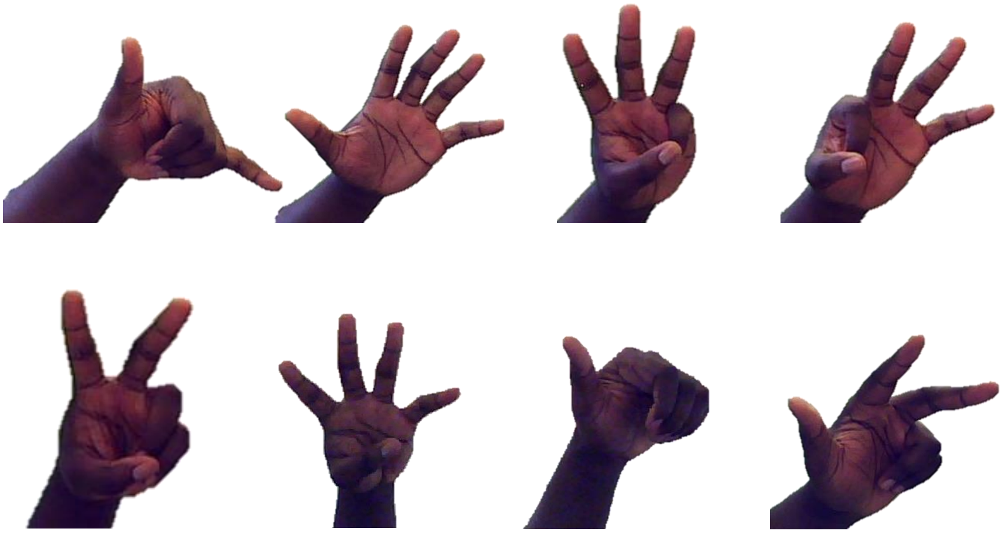


Figure 3.17 The six different poses captured in Dataset B.

cheeseboard calibration between the reference stereo image and the RGBD camera (see Section 3.1.3 for more details). 12 participants partook in the data collection with a total of 6,000 (500 per participant) captures collected, each articulating eight poses. Like the first dataset, the problem that was to be addressed with this dataset is hand depth from stereo. Hence each data instance consists of a pair of stereo (left and right) and depth capture data of the hand.

3.3.3 Dataset C

The final dataset contains a similar variety of hand poses to Dataset B. However, in addition to the stereo images and RGBD capture, the location of hand joints are captured. The reader is directed to Section 3.5 for more detail on the hand pose model used. It also consists of the same number of participants, where a total of 12,000 (1,000 per participant) captures were made. It was used to validate pose estimation from stereo captures. Since the proposed technique to be validated uses depth as a hidden variable, depth was also collected. Hence, each data instance consists of a triplet of captures namely: stereo, depth and hand pose.

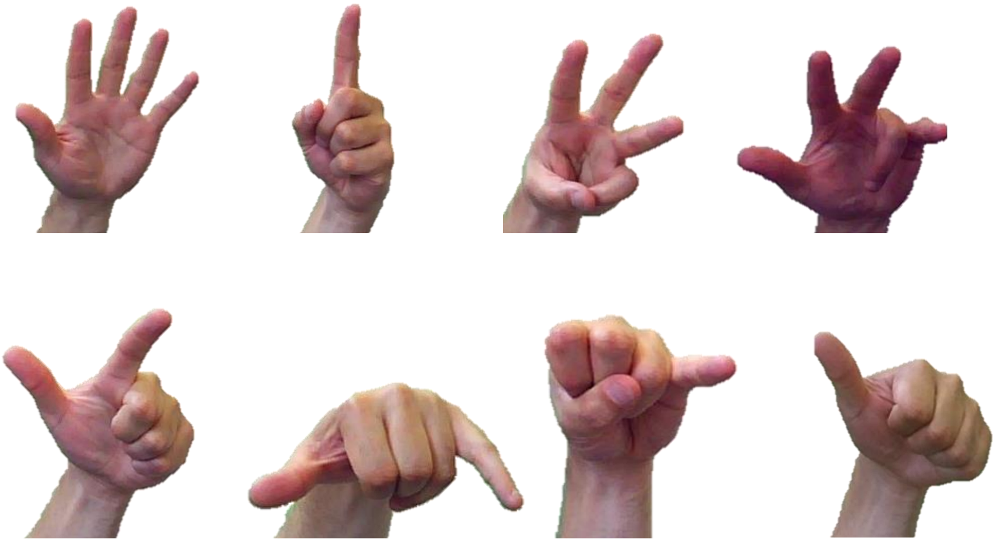


Figure 3.18 The six different poses captured in Dataset C.

To establish a database of strong registration between the triplet of data: stereo, depth and pose acquisition was carried out on the stereo camera, a RGBD camera, and an off-the-shelf hand pose detector. The registration between the RGBD and stereo remains the same as was in Dataset B. It suffices that the spatial position of the hand pose detector relative to the stereo camera is unchanged during the capture of training data. The result of this is a well registered stereo-depth image pairs along with a pose vector. The stereo-RGBD calibration was done with a re-projection error of $11.564mm$. See Section 3.4 for details on the hardware used for collecting data.

An overview of all three datasets is presented in Table 3.1.

Dataset	**Dataset A**	**Dataset B**	**Dataset C**
Number of participants	5	12	12
Number of samples	1,000	6,000	12,000
Number of poses	6	8	8
Domains	Stereo and depth	Stereo and depth	Stereo, depth, and pose
Chapter used	4	5	6 and 7
View point	Only fronto-parallel view	Arbitrarily oriented view	Arbitrarily oriented view

Table 3.1 A comparative overview of all three datasets used to train and validate the work presented in the latter part of this thesis.

3.4 Hardware Used

The datasets introduced above consisted of three main domains, namely: stereo, depth, and pose. To capture data in each of this domains, three devices were used. These include a stereo camera, an RGBD camera, and a hand pose estimation device.

The stereo camera that was used is the Minoru 3D Webcam[3]. The Kinect Sensor for Xbox One[4] was used as an RGBD camera to capture depth whilst the hand pose estimation device used was the Leap Motion Controller[5].

3.5 Hand Model

The hand model used to model hand articulation in this thesis consists of 20 joints. These included the wrist; the thumb (fingertip, distal and intermediate); the index, middle, ring and pinky finger (each with a fingertip, distal, intermediate and proximal joint) as shown in Figure 3.20. Each of these 20 joints has three variables (including its X, Y and Z coordinate), yielding 60 degrees of freedom. There is no kinematic constraint on the model's joint angles, however, a hand prior model was computed based on a dataset of poses (see Chapter 6 for more detail).

Figure 3.19 (a) The Minoru 3D Webcam (b) The Kinect Sensor for Xbox One (c) The Leap Motion Controller

[3] http://www.minoru3d.com/
[4] https://www.xbox.com/en-GB/xbox-one/accessories/kinect
[5] http://store-eur.leapmotion.com/products/leap-motion-controller

3.6 Summary

The camera model and machine learning concepts on which this thesis is based have been presented. The chapter begins with the introduction of the pinhole camera model before looking at multi-view geometry which provided a basis for solving the problem of registering RGBD camera to a stereo camera. The RGBD-stereo camera registration is preceded with camera calibration to determining the extrinsic and intrinsic parameter of the stereo rig. The second part of the chapter delved into the machine learning that informs the processes of inferring higher level information from recovered and processed information from the camera. This entails a brief introduction to probability modeling; learning and inference; and the significance of uncertainty in the learning procedure. This was used to introduce the four main machine learning frameworks that were used in the thesis namely: Random Forest, conditional random field, convolutional neural network, and Markov-chain Monte Carlo. The next chapter presents the first

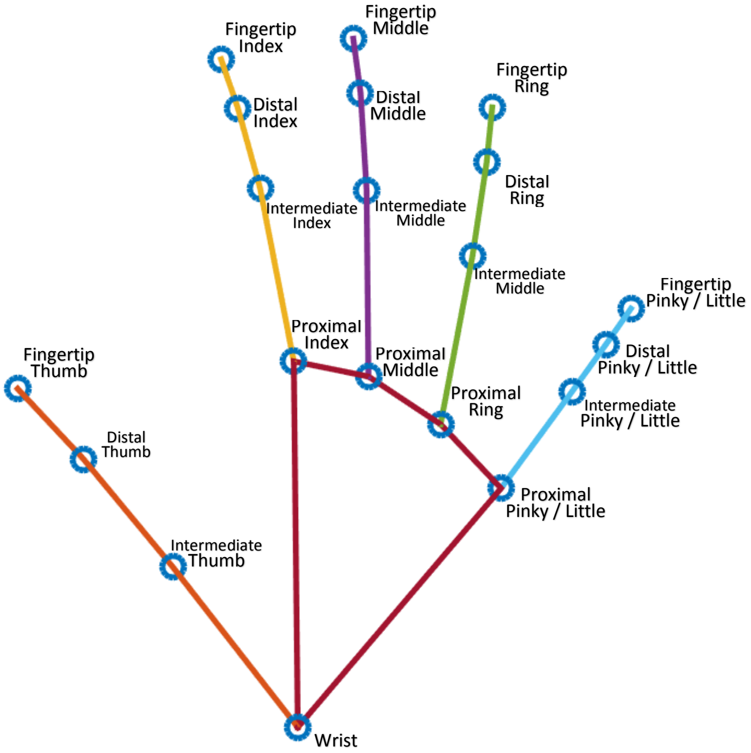


Figure 3.20 An illustration of the hand model used in this thesis.

proposed framework in the thesis. It proposes a novel Eigen-based variant of Regressive Random Forest with the aim of establishing a mapping between low-quality disparity to high-quality depth.

Chapter 4

Robust Hand Depth Estimation using Eigen-based Regression Forest

As introduced in Chapter 2, there are inherent limitations to the recovery of disparity by correspondence mapping, such as a large search space, textureless regions, and occluded regions. These limitations make the recovery of an accurate depth map from a stereo image pair a very challenging task. This is especially true when the recovered depth is to be used in the complex task of estimating hand articulation. Clearly, there is a need for a framework that is able to use low-quality disparity to reconstruct more robust depth information of viewed scenery. Human vision, which is able to efficiently discern articulation and perform tracking activities, merely from stereoscopically recovered depth, uses the brain in processing disparity recovered combined with previous experience. In this chapter, the aim is to use a machine learning regressive-based mapping framework to improve a low-quality disparity image. The first task is to retrieve the disparity image from a stereo image of a scene with a hand pose. This is then used to extract a high-quality depth image using a Regressive Random Forest. More precisely, given a stereo recovered disparity image the aim is to map this to a high-quality depth image that can later be used for estimating hand articulation. The proposed technique relies on a robust hand segmentation procedure. However, given the long history of this topic in computer vision, it is not addressed. There is a large body of work on the topic (see [100–102]). This chapter does not elaborate much on the conventional concept of Random Forests; rather interested readers are referred to Section 3.2.4 for more information.

Contributions: This chapter proposes the application of a novel, data-driven Regressive Random Forest framework that learns the mapping between a lower quality disparity estimation and high-quality groundtruth depth measurement.

It presents a novel variant to regression forests and the concept of Eigen Leaf Node Features (ELNF) is proposed. ELNF factorizes for the posterior probability and regresses the depth using highly discriminative features. It should be noted that ELNF has much wider application, and it can be used in other regression problems outside the context of depth recovery. The ability of ELNF to more accurately estimate the depth of hands compared to conventional Random Forest regression is also explored.

4.1 Disparity Estimation

The first stage in the pipeline involves the recovery of the disparity from the stereo image pair, where the key challenge is to identify correspondences between the views. In the context of this application (stereo depth recovery of a hand), three main issues arose: (i) the search space is large; (ii) there is inconsistency in the cameras used to image the scene; and (iii) some image regions are textureless. In order to address the issues of search space size and textureless regions, the stereo rig is calibrated using stereo camera calibration [103]. The epipolar geometry of the stereo rig is computed in the form of the fundamental matrix. This is in turn used in estimating the epipolar line. So, given a pixel (of a hand region) in the first stereo image, the search for the corresponding point on the other image is carried out on this epipolar line (see Section 3.1 for more detail). Note that alternatively, rectification could be applied to maintain parallel epipolar lines, allowing for strictly horizontal shift in the search for correspondence.

Rectification was not used as it yielded slightly misaligned epipolar lines as a result of warping the image. Although this misalignment was minor, it was not negligible to the stereo matching process of the skin-based scene. A further constraint is imposed, prior to the stereo-matching procedure, by applying skin region segmentation which restricts the search space of correspondences. The lack of consistency between the cameras remains an issue here. Hence, a previously proposed cost function, Quantized Census, which has been experimentally proven to be able to compensate for various types of radiometric difference in the stereo image pair, is used. Interested reader is referred to Appendix 1 for more details on Quantized Census. Typical disparity images recovered from stereo captures using Quantized Census are presented in the top row of Figure 4.1, along with their corresponding groundtruth depth in the bottom row. The disparity images are measured in pixel shifts, the brighter the color the larger the pixel shift at a given pixel. On the other hand, the depth images are measured in millimeters (mm), the hotter the color the closer the pixel is to the camera. Observe that some depth cues are discernible (e.g. the fingers sticking out) from the disparity, however, there are still lots of artifacts (e.g. the wrist region). The purpose of regression framework that is presented in the following section is to improve the quality of the depth information contained in the disparity images.

4.2 Mapping Disparity to Depth

The task here is to establish the mapping, $I_{disp}(\boldsymbol{x}) \rightarrow I_{depth}(\boldsymbol{x})$, at a pixel position $\boldsymbol{x}$, between the disparity image and the groundtruth depth image from the RGBD sensor. This mapping is modeled with a Regressive Random Forest based on Kernel Density Estimation. The learning procedure attempts to simplify the mapping by first classifying it into subspaces, where each subspace is a quantized range of depth values. This is inherently embodied in the conventional classification Random Forest framework. This will then allow for an efficient and finer regression at leaf nodes.

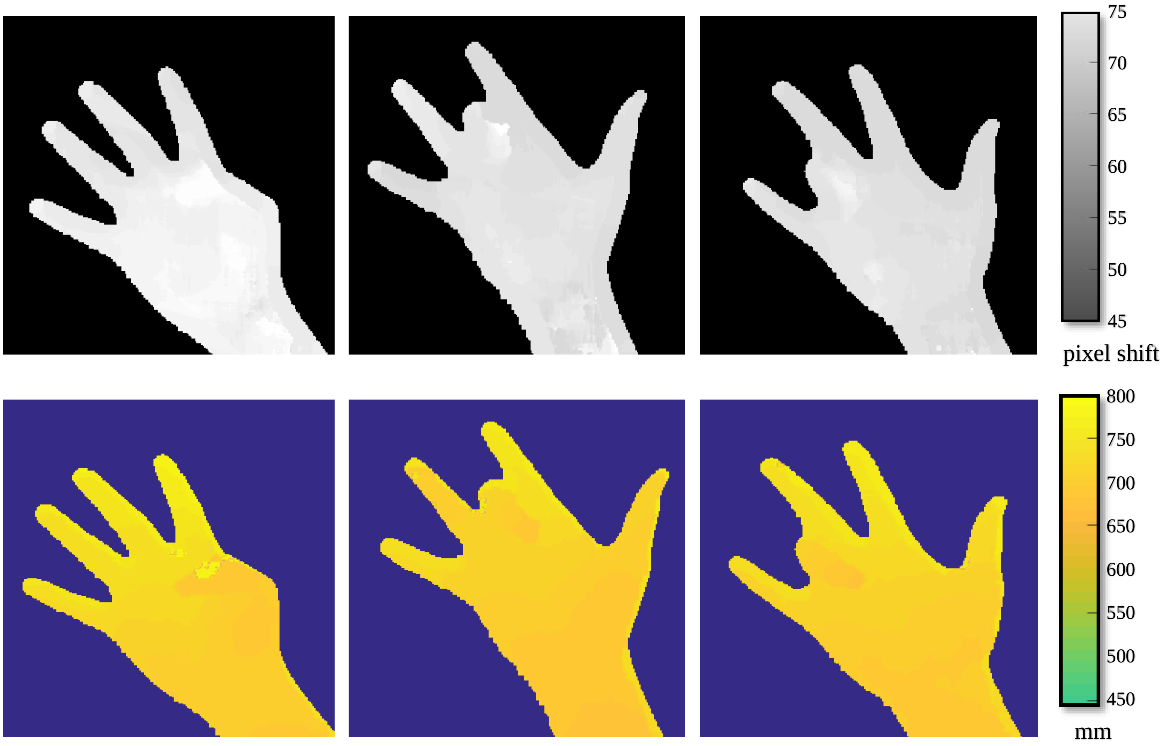


Figure 4.1 A comparison of disparity image and groundtruth depth. The top rows show some recovered disparity images from stereo captures of hands. The corresponding groundtruth depth images are displayed in the second row. Observe that some depth cues are discernible (e.g. the fingers sticking out) from the disparity, however, there are still lots of artifacts (e.g. the wrist region).

The prediction of the depth at a given pixel position is made based on a feature generated from two vectors as introduced in Section 2.2.2. Here the majority disparity position in the hand region d_M, is used as the normalizing factor in place of the depth, as in

$$f_\theta(I_{disp}, \boldsymbol{x}) = I_{disp}\Big(\boldsymbol{x} + \frac{\boldsymbol{u}}{d_M}\Big) - I_{disp}\Big(\boldsymbol{x} + \frac{\boldsymbol{v}}{d_M}\Big). \tag{4.1}$$

Recall from Section 2.2.2 that $\boldsymbol{u}$ and $\boldsymbol{v}$ are a pair of random offset vectors applied to the pixel location, $\boldsymbol{x}$. Henceforth the pair of vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$, will be referred to as feature vectors, $\boldsymbol{\theta} = \boldsymbol{u}, \boldsymbol{v}$. Note here that the feature vector in this case is applied to disparity image as opposed to the depth image (as introduced in Section 2.2.2).

4.2.1 Random Forest

As previously stated, the Random Forest learner presented in Section 3.2.4 was used. N decision trees are grown by recursively splitting and passing training data, S, into two sub-nodes S_i. To increase the variation in the different feature levels that can be generated, a Gaussian filter is applied to the highly discrete valued disparity image due to pixel shift based values. As in the work of Criminisi and Shotton, randomness is maintained in the bagging, [104], feature selection and threshold selection; and the tree aims to decrease the entropy of the training dataset by maximizing the information gain.

$$I_G(\theta) = E(S) - \sum_{i\epsilon\{L,R\}}^{n} \frac{|S_i(\theta)|}{|S|} E(S_i(\theta)). \tag{4.2}$$

Entropy is defined as

$$E(S) = log(\sigma_s), \tag{4.3}$$

where σ_s, is the standard deviation of the depth values of the pixel points within the subset, S. Statistical analysis is carried out on the pixels that land at each leaf node. The distribution of the features computed against the actual depth is established (Figures 4.2a and 4.2d). Recall that for a single pixel position, an infinite amount of vector features could be generated, by randomly selecting any amount of offset vectors. It would be impractical and redundant to use all these features. Hence a subset of these features is used to establish the relationship between features and groundtruth depth. ELNF is proposed to determine this subset of features, and this is described in Section 4.2.2. With the subset of features in place, multivariate Kernel Density Estimation is applied by convolving the features-actual depth distribution with a Gaussian kernel [105]. For a subset of N features, this yields a continuous $(N + 1)$ dimensional distribution of the feature(s) against the actual depth. Figures 4.2b and 4.2e show the resulting distribution when $N = 1$, i.e. the number of features used is one. Here the frequency of this distribution is represented in the third dimension of the plot. The resulting continuous distribution is stored at the leaf node to be evaluated during testing (Figures 4.2c and 4.2f). Determining the features to use will be elaborated upon in the following section.

Forest predictions: Each pixel point, x, whose depth is to be predicted, is passed through each of the trees in the forest ensemble. At each node the pre-established splitting function, $f(\theta, \phi)\{L_n, R_n\}$ is evaluated based on the feature and the pre-established threshold, ϕ. Determining whether to send the pixel to the left, L_n, or to the right node, R_n. This is repeated recursively until the leaf node is reached. At this point, the feature (computed using pre-specified vector pair discussed in Section 4.2) is used in factorizing for the posterior probability, $Pr(d|\theta)$, of depth, d, given that the

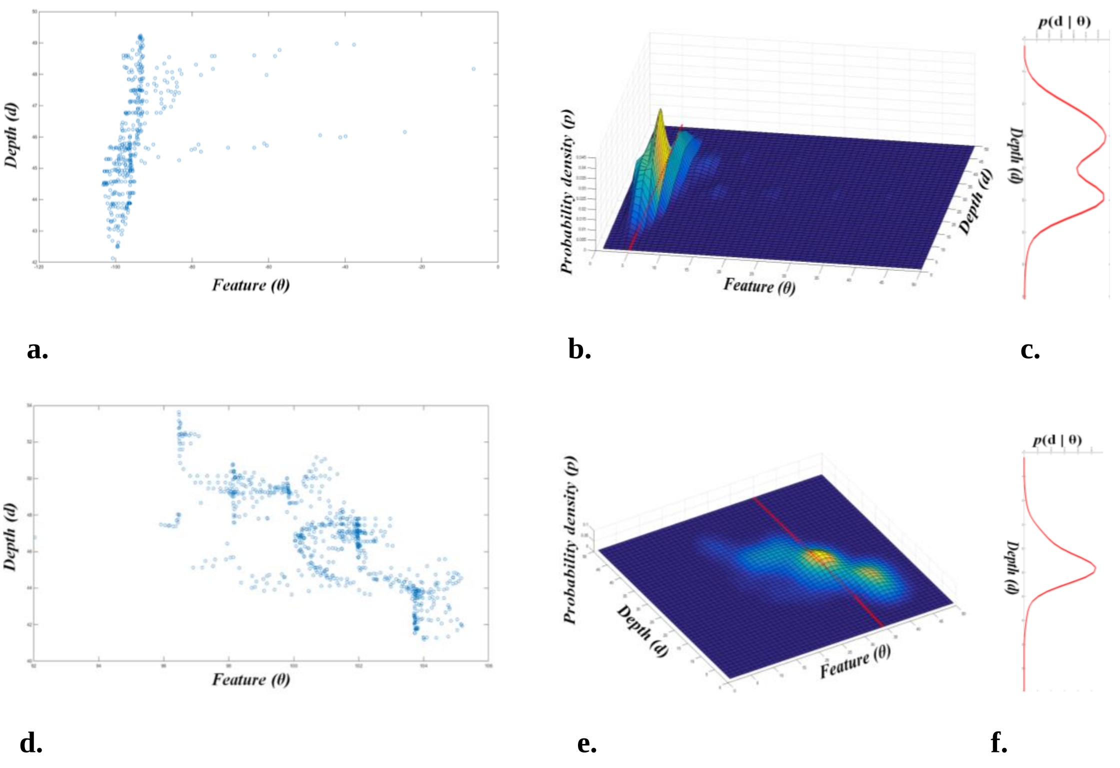


Figure 4.2 At a leaf node, a depth-feature distribution is established (a). Images (a, b & c) illustrate a bad feature-depth distribution (vertically orientated). In contrast, (d, e & f) illustrate a better feature-depth distribution (obliquely orientated) as factorizing yields a confident posterior (f). ELNF is biased towards the obliquely orientated distribution.

pixel point has a feature, θ, as in (Figure 4.2c and 4.2f). This probability is aggregated across the ensemble of trees, t.

$$Pr(d|x) = \frac{1}{N} \sum Pr_t(d|\theta) \tag{4.4}$$

Note again that in the Figures 4.2b and 4.2e, the number of features is just one, computed from a single learned vector pair. However, improved results are achieved from using two features (i.e. $N = 2$), see Section 4.3. Due to computational limitations, experiments with more than two features were not tested due to the fact that the computational cost of the multi-dimensional Kernel Density Estimation and the size of its resulting distribution increases exponentially with increasing dimension.

4.2.2 Determining Eigen Leaf Node Features (ELNF)

The problem with factorizing for the posterior probability, $Pr(d|\theta)$, is that the distribution might not have a strong correlation between the feature and the depth to be estimated. Consider the Figures 4.2d and 4.2e. They convey a strong negative correlation, hence factorizing for the posterior probability of the depth yields a small standard deviation and subsequently more confidence in predictions is made by maximum likelihood (Figure 4.2f). In contrast, the distribution in figures 4.2a and 4.2b exhibits a weak correlation. Here the factorized posterior yields less confidence (Figure 4.2c). As each pixel position at the leaf node has potentially infinite features, it would be efficient to select those that are most discriminative for regression.

The task, then, is to ensure that feature(s) selected at the leaf node will yield a strong positive or negative correlation. To establish this, the principal eigenvector and the ratio of the two eigenvalues of the covariance matrix of the distribution are exploited, using what is coined as Eigen Leaf Node Features (ELNF). In this case, it is useful to establish an obliquely orientated distribution (Figure 4.2a) as opposed to a vertically orientated principal distribution (Figure 4.2b). The ratio of the two eigenvalues represents how compact the distribution is in the principal direction relative to the perpendicular direction. Hence at the leaf node, the feature, θ that minimizes the following cost function is selected:

$$E(I_{depth}(\boldsymbol{x}), \boldsymbol{f}_\theta) = \alpha(|\Delta(v_1)| - 1)^2 + (1-\alpha)\frac{\lambda_2}{\lambda_1} \tag{4.5}$$

where $\Delta(v_1)$, is the slope of the principal eigenvector, v_1, of the covariance matrix of the distribution of the actual depth, $I_{depth}(\boldsymbol{x})$, and the feature, f_θ, for a pixel point, x, in the leaf node. λ_1 and λ_2 are the two eigenvalues, the former being the principal eigenvalue. $\alpha \in [0, 1]$ is a weighting providing a convex combination of the terms. The first component of equation,$(|\Delta(v_1)| - 1)^2$, quantifies how close the magnitude of the slope is to 1 whilst the second component, $\frac{\lambda_2}{\lambda_1}$, quantifies the spread in the direction orthogonal to the principal eigenvector. Consequently, maximizing $E(I_{depth}(\boldsymbol{x}), \boldsymbol{f}_\theta)$ will encourage a more obliquely oriented principal component of distribution and increase compactness. The more compact the posterior prediction, the stronger the accuracy of the recovered depth, which in turn improves the performance of potential pose estimation that could follow.

When the number of features selected at the leaf node is more than one (i.e. the number of dimensions of the resulting groundtruth depth-features distribution becomes more than two), the aim becomes to maximize the dependency between all possible pairs of these dimensions. Subsequently, Eq. 4.5 is applied to the distribution of all pairs of either features or groundtruth depth across all pixels arriving at a particular leaf node. Hence,

$$E(I_{depth}(\boldsymbol{x}), \boldsymbol{f}_\theta) = \sum_{n,p} C(n,p), \forall n, p = 1, ..., N+1 | n \neq p \tag{4.6}$$

where

$$C(n,p) = \alpha(|\Delta(\boldsymbol{v}_{1,n,p})| - 1)^2 + (1-\alpha)\frac{\lambda_{2,n,p}}{\lambda_{1,n,p}}. \tag{4.7}$$

$\Delta(\boldsymbol{v}_{1,n,p})$ is the slope of the principal eigenvector, $\boldsymbol{v}_{1,n,p}$, that corresponds to the distribution of the n^{th} and p^{th} columns of a data matrix, $\boldsymbol{D}$.

$$\boldsymbol{D} = [\boldsymbol{d}_x | \boldsymbol{f}_\theta^1, \boldsymbol{f}_\theta^2, ..., \boldsymbol{f}_\theta^N] \tag{4.8}$$

Here, $\boldsymbol{D}$ is the resulting matrix when the groundtruth depth vector, $\boldsymbol{d}_x$ (consisting of the depth of all pixels at the leaf node), is concatenated with the features matrix (consisting of the feature values computed at these pixel locations). During implementation, α was set to 0.7 based on experimental performance. Distributions that minimize the cost function, E, in Eq. 4.6 and 4.5 are those for which the principal orientation has a slope of closer to 1 or -1 and greater compactness along the principal orientation.

4.2.3 Training

Each tree in the ensemble is trained individually, using random samples of subsets of the training dataset, S. The recursive splitting procedure described in Section 4.2.1 is executed. The partitioning of the dataset stops when the maximum level of depth; minimum entropy in the dataset; or a minimum number of data samples in the dataset is reached. Note that all vector features are extracted from the disparity images, I_{disp}, while the associated output depth is retrieved from the groundtruth depth image, I_{depth} (acquired from the RGBD sensor). At a leaf node, random feature vectors are generated, and the one that minimizes the energy of Eq. 4.6 is selected as in:

$$\boldsymbol{\theta} = argminE(I_{depth}(\boldsymbol{x}), \boldsymbol{f}_\theta). \tag{4.9}$$

Dataset

To demonstrate the proposed technique, training was carried out on both real and synthetically generated data.

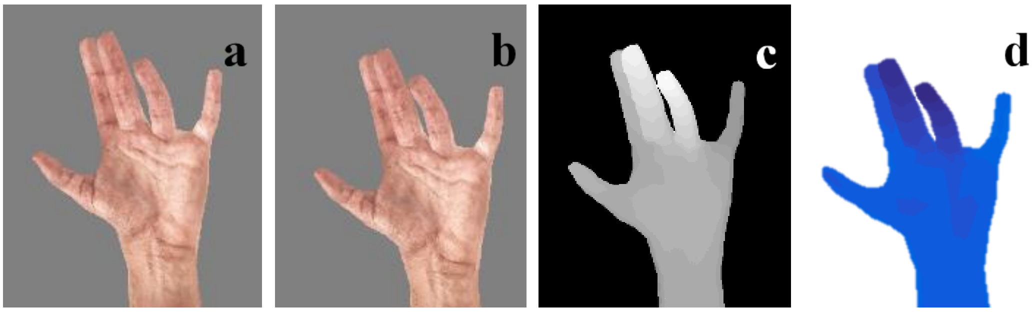


Figure 4.3 Rendered RGB images of the synthetically articulated hand from two perspectives (a & b) and the resulting disparity (c). The lighter the colour the larger the pixel shift. The corresponding groundtruth depth (d) is extracted from the depth buffer. The bluer the colour the closer the pixel is from the synthetic camera.

Real Data: Dataset A presented in Section 3.3.1 was used. The consisted of hand captures from five participants, a total of 1,000 (200 per participant) captures were made, each articulating seven poses. Variation in poses consisted of simple finger flexion and extension with the relative hand to camera orientation kept the same at the fronto-parallel view.

Synthetic Data: The synthetic dataset was produced using computer-generated hand poses. A rigged hand model from [106] was skinned and rendered using an OpenGL implementation and produced images from two perspectives by horizontally displacing position and the distorting orientation of the synthetic camera (simulating a non-parallel stereo camera pair as in Figure 4.3a and 4.3b). The image pair was passed into a stereo-matching framework as was the case with real data using the known rotation and translation information as well as the perspective projection of the camera to compute a fundamental matrix. The disparity estimation procedure explained in Section 4.1 is applied (Figure 4.3c). The depth buffer was also read at the same camera position as the reference stereo image pair (Figure 4.3d), establishing direct registration. A dataset consisting of 10,000 instances of different articulations was generated. This was achieved by adjusting the different joint angles of a skeleton which moved the skinned mesh. See Figure 4.4 for some sample captures.

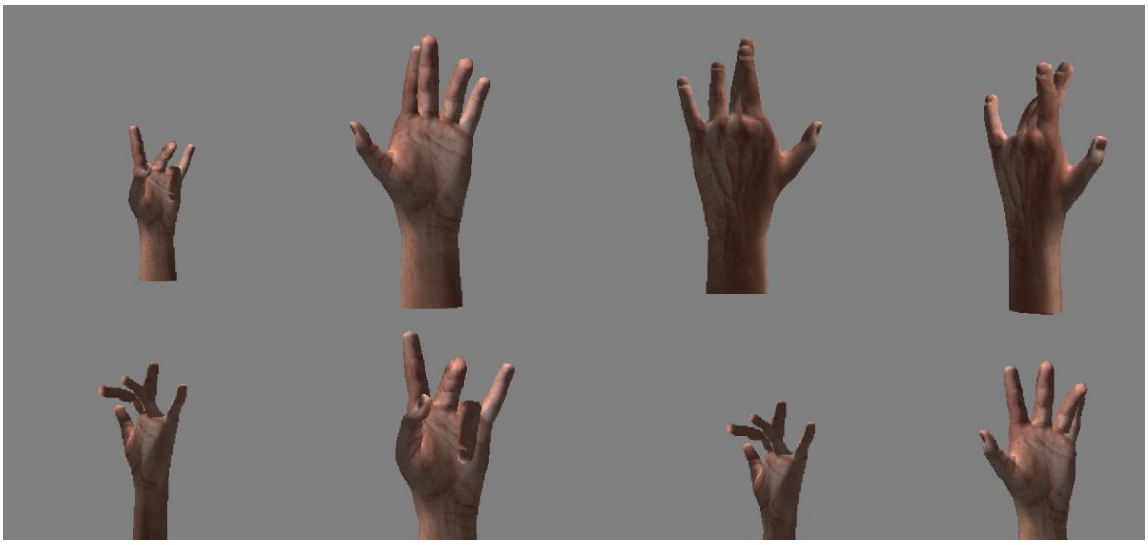


Figure 4.4 Some examples illustrating the variety of articulations in the synthetically generated dataset.

Implementation details: The random forest was implemented, trained and tested on a Quad core i7 processor CPU. At test time, based on the MATLAB implementation, the stereo matching algorithm that preceded runs in 0.43 seconds on average to recover the disparity map. Each test-time forward pass through a forest of 80 trees takes typically 0.019 seconds at a maximum depth level of 12. Considering that a typical hand region consists of 21,000 pixels, a typical forward test typically took 440 seconds. It should be noted each forward pass through a tree is a simple set of operation and hence real-time implementation can be achieved by GPU implementation (similar to the work of Shotton et al. in [65]).

4.3 Experiments

To demonstrate the technique, training was carried out on both real and synthetically generated data. The approach was experimentally validated, presenting both qualitative

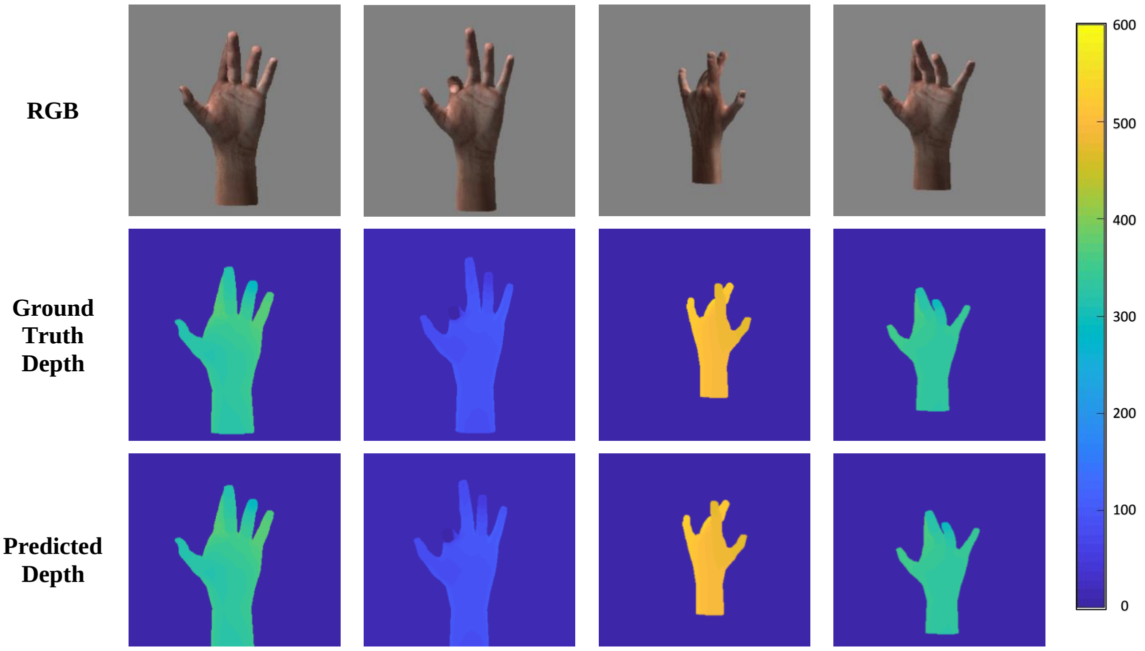


Figure 4.5 Qualitative results using synthetic captured data on four instances of poses. The 1^{st} row are reference RGB synthetic hand poses. The 2^{nd} and 3^{rd} row presents the ground truth and predicted depth as a result of using the proposed method. Observe how almost a perfect recovery was achieved in the synthetic scene.

(Figure 4.6) and quantitative (Figures 4.8, 4.9 and 4.7) results. The synthetic dataset yielded a highly accurate result as shown qualitatively in Figure 4.5.

Experiments were carried out with the aim of exploring: (i) the significance of ELNF, (ii) the significance of the disparity information, and (iii) the effect of using multiple leaf node features. The experiments will be discussed in greater depth in the following sub-sections. The results were quantitatively appraised by taking the average absolute difference between the actual depth, d_{GT} and the predicted depth, d_p across all hand region pixels, $\frac{|d_{GT}-d_p|}{N}$, where N is the number of pixels in the hand region.

4.3.1 Evaluating for the Significance of ELNF

In this section, the significance of ELNF is investigated experimentally. The dataset was trained and tested with the conventional regression forest and then on an implementation of Random Forest that is augmented with ELNF. A qualitative comparison

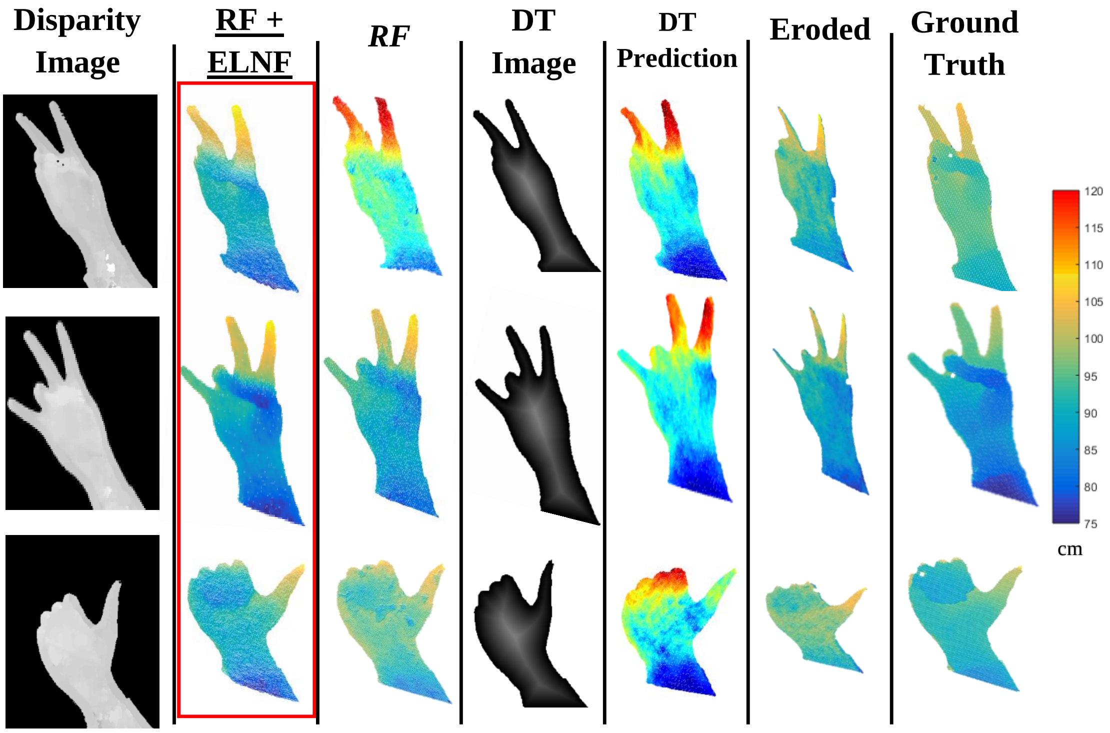


Figure 4.6 Qualitative Results using real captured data on three instances of poses. The 1^{st} and 4^{th} columns are input to the Distance Transform-based prediction and to the Disparity based prediction respectively. The 2^{nd} and 3^{rd} columns are predicted depth from the proposed framework with and without using ELNF respectively, while the 5^{th} and 6^{th} columns show predicted depth using ELNF on distance transformed images and eroded disparity images respectively. The final column shows the groundtruth depth from the RGBD Sensor.

is presented in the 2^{nd} and 3^{rd} columns of Figure 4.6, and more quantitative results in Figure 4.8.

Qualitative analysis: Examining the 2^{nd} and 3^{rd} columns of Figure 4.6, a substantial improvement in the predicted depth and overall shape of the hand can be seen when ELNF is used. In all cases, it is clear that ELNF predicts a stronger holistic hand shape compared to RF; the digits are more discernible for instance. This is due to the stronger correlation enforced between the feature and the output depth at the leaf node distribution, owing to these specialized features.

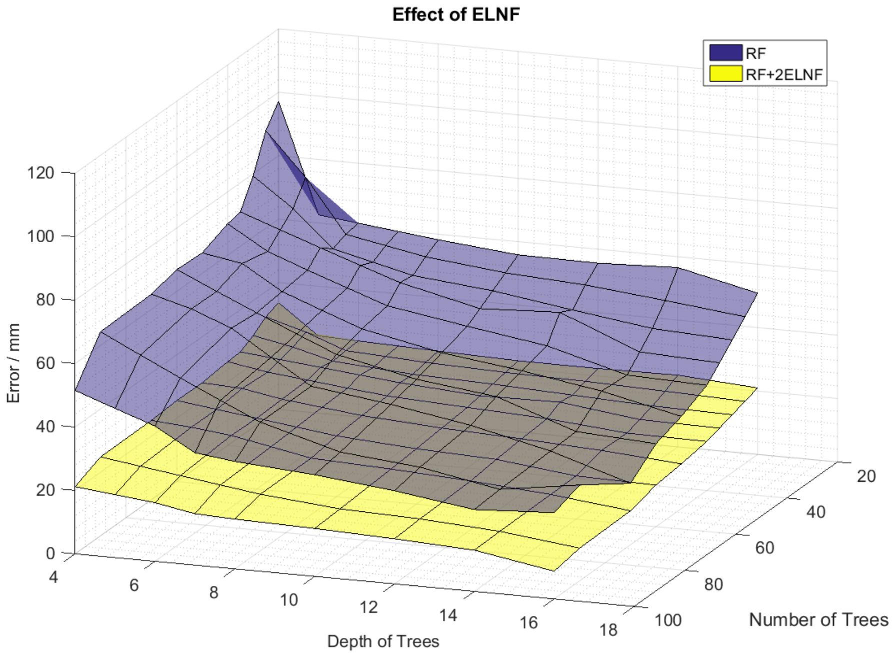


Figure 4.7 Quantitative results showing error in depth prediction at different depth of trees and number of trees. Here the results of a conventional Random Forest are compared to one augmented with ELNF. The lower the position of the surface plot the better the performance. The 2ELNF is in the legend of the plot indicates that two features were evaluated at the leaf node.

Quantitative Analysis: The omission of ELNF had an interesting result. At low tree depth and small number of trees, the significance of ELNF is more apparent. This is due to the fact that at low tree depth and small number of trees, the entropy at the leaf nodes is high, and ELNF implicitly introduces less entropy when the distribution is factorized based on the pixel feature. The superiority in performance provided by ELNF is reduced as the tree depth is increased. This is due to the fact that the entropy of the data reaching the leaf node decreased due to the lack of a strong correlation between the feature and the depth.

4.3.2 Comparison with disparity input to distance transform input

An early concern was that the successful prediction of depth was mainly dependent on the contour of the hand (that is acquired from the initial hand region segmentation) and not on the disparity image. This would render the stereo-matching step insignificant and hence a single camera setup with hand region segmentation would be sufficient. Two experiments were carried out to address this issue. First, the proposed framework was trained and tested with an image generated solely from hand region segmentation of a single view image instead of a disparity image. A distance transform of the hand region segmented image was used, because the vector feature used relies on a continuously varying intensity image (4^{th} column of Figure 4.6). Secondly, to further investigate to what extent the proposed framework depends on the segmentation pre-step, a test was carried out by applying the proposed framework on instances of eroded segmentation of the disparity images. The consequence of this is the lack of shape information, hence depth prediction will be solely based on disparity information.

Qualitative Analysis: Figure 4.6 (in the 5^{th} column) clearly illustrates qualitatively the significance of the initial depth information provided by disparity input. The distance transformed input only contains the contour of the segmented hand region. This significance is highlighted particularly in regions where the entire shape of each finger is not discernible from the contour, for instance in the distal end of the index finger in the second row. On the other hand, the result from the eroded disparity looks more promising (6^{th} column). Whilst this is less visually correct in comparison to the properly shaped hand, one can still discern part of the bent finger, for instance.

Quantitative Analysis: The quantitative results in Figure 4.8 reflect the previously described qualitative result, in that the average error in using disparity is $21.76mm$ in comparison to $71.76mm$ when a distance transform is used. This clearly illustrates the significance of depth information from the disparity image. The distance transformed input and disparity input are affected similarly as the depth of the tree increases. The tree depth signifies how specialized the tree is to the dataset. The randomness in and the ensemble nature of random forest increases the prediction variance. Whilst the increase in the number of trees does not increase generalization error (i.e. the error when the random forest is tested on unseen data), the depth of each tree can. This is because the random forest is still susceptible to the potential difference in the distribution of data points in the feature space of the training dataset to that of the testing dataset. As well as this, the effect of outlier points in the training dataset is magnified in the deeper trees. As discussed earlier, a decision tree can be thought of as the partitioning of a prediction problem into simpler subspaces (in the leaf node) that can then be modeled with simple models like a Gaussian. Deeper split levels yield more partitions and subsequently fewer data points per subspace (at each leaf node). Consequently, the negative influence of outlier points at leaf nodes is magnified as due to the small population of non-outlier points. As a result of these two factors, the

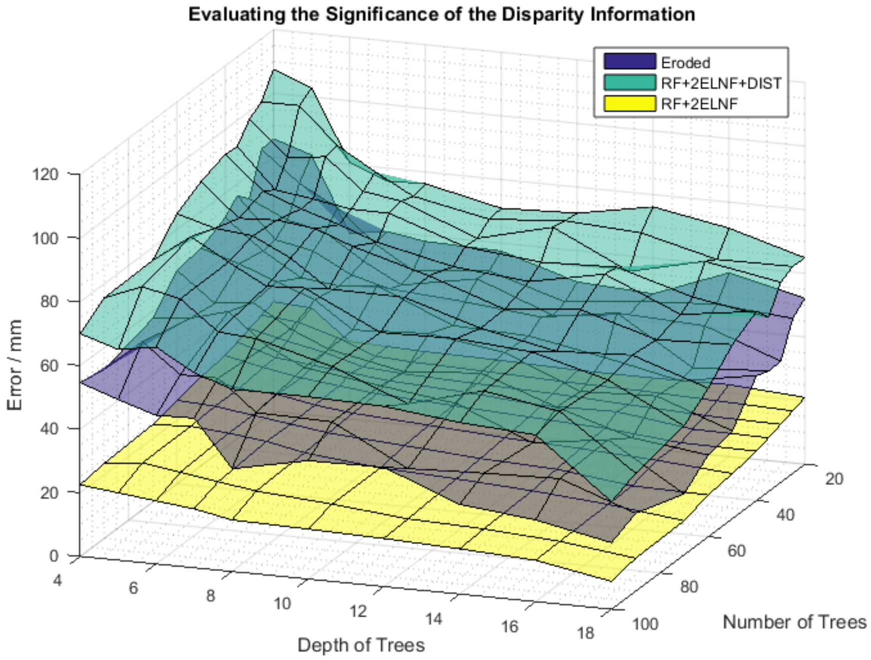


Figure 4.8 Quantitative results showing error in depth prediction at different depths of trees and number of trees. The significance of disparity information is investigated by predicting depth from a distance transform input and from a disparity where the edges of its segmentation eroded.

forest fails at predicting unseen data when it becomes too specialized (due to deeper trees).

4.3.3 Single leaf node features vs two leaf node features

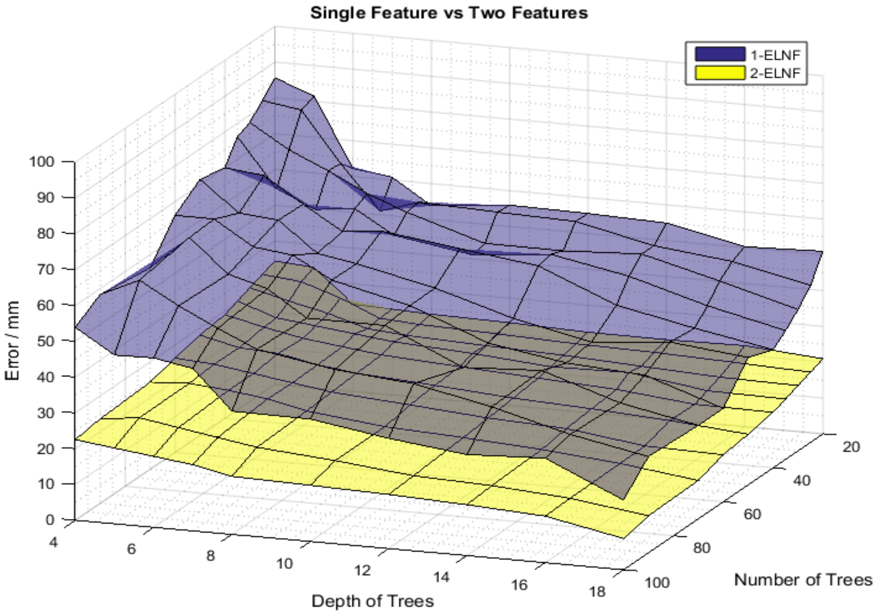


Figure 4.9 Quantitative result showing error in depth prediction at different depth of trees and number of trees. The significance of using multiple features is evaluated by comparing single feature ELNF to two feature ELNF.

To investigate the effect of multiple leaf node features, predictions from trees built with a single leaf node feature are compared to those built with two (Figure 4.9). Note the significant decrease in error along different tree depth and number of trees as a result of using two features. This improvement is more apparent at lower number of trees and depth. This is due to the fact that the extra features helps to further discriminate between different depth levels.

4.4 Summary

In this chapter, an innovative application of the regression forest technique for upgrading disparity to depth was proposed and developed. The disparity image was first retrieved using greedy search stereo-matching. A Regressive Random Forest was then applied to learn the mapping from the disparity information (and hand shape context) to hand depth. The regression forest used in this case is unique due to the use of Eigen-based feature selection. In doing this, a cost function was proposed that estimates for ELNF that is more suitable for regression at training. The consequence of this work chapter is a proof of concept that the low-quality disparity can be improved upon to generate a more robust high-quality depth estimate using a machine learning approach. This novel approach is a variant of the Regressive Random Forest framework and is applicable beyond the context of the core aim of this research.

Although some promising results were achieved, the dataset set used is limited in terms of the fact that it only entailed fronto-parallel view poses. Also, the performance of the proposed approach in this chapter is still dependent on the stereo-matching results that precedes it. Lastly, the mean depth error of $21.76mm$ is limiting, as, within such margin of error, the fingers (typically $16 - 20mm$) close to each other might not discernible. Seeing as this is an objective of the thesis to use the estimated depth to resolve hand pose, clearly, there is a need for improvement. The following chapter introduces a more robust solution, that incorporates the affinity costing into the machine learning framework. Instead of pre-computing disparity (which can be erroneous) as proposed in this chapter, the matching pixel decision is implicitly determined in the machine learning framework.

Chapter 5

Stereo-based Hand Depth Recovery using CRRF

In the previous chapter, a novel approach to hand depth estimation from stereoscopic images was presented. In this chapter, the work on high accuracy depth recovery is continued. Specifically, a more robust and advanced method to that introduced in the previous chapter is proposed. Similar to the previous technique, the proposed approach remains a data-driven one, however here a superpixel-based [1] regression framework is presented that takes advantage of the smoothness of the hand's depth surface. To this end, a novel method that combines a closed-form Conditional Random Field with learned weights and a Regressive Random Forest (RRF) with adaptively selected expert trees is introduced to model the mapping from a stereo RGB image pair directly to a depth image. Note the removal of the stereo-matching pre-step. The intuition behind the proposed RRF is that it adaptively selects different stereo-matching measures as it implicitly determines matching pixels in a coarse-to-fine manner. While the RRF makes depth prediction for each superpixel independently, the CRF unifies the prediction of depth by modeling pair-wise interactions between adjacent superpixels. As a result, the proposed system provides a robust method for generating a depth image with an

[1] A superpixel is defined as a contiguous cluster of pixels. The cluster is based on the relative proximity and intensity values of the pixels within the cluster.

inexpensive stereo camera. The latter part of the chapter presents both qualitative and quantitative results, that demonstrate the superiority of this approach to that in the previous chapter.

The goal remains to extract robust hand depth from the stereo RGB inputs as a precursor to hand pose estimation. Whilst recovery of hand depth provides challenges, as previously expressed in prior chapters, the constraint that the depth recovery task will only apply to a particular class of object (hand) means that stereo-matching constraints can be learned using a machine learning approach and tested on similar surfaces. This is particularly useful as one can better establish the matching criteria that can achieve the best stereo matches (and hence disparity) since the typical structure of the "scene" of interest is known. Conventional approaches to stereo-matching (like the one used in the previous chapter) rely on universal conditions for finding correspondences. Specific to the problem of hand-based stereo-matching, a more robust approach is proposed that adaptively establishes matching conditions based on unique properties of the hand (e.g., skin tone, texture, etc.). Underlying this approach are four main conjectures. The first is that the depth surface of a scene with hand poses consists of a set of homogeneous regions that yield a smooth surface and continuous texture. Second, for establishing correspondences, particularly in the presence of ambiguities like textureless regions, a diverse set of matching costs and window sizes improves the chances of finding a correct match. Using different matching criteria to assess potential matches effectively increases the dimension of the feature space that is used to determine similarity. This is particularly the case with attempting to establish correspondence on an inherently untextured hand region. Third, that the difference in skin tone and hand size for different individuals makes establishing universal matching criteria for determining stereo correspondence a difficult task. Conventional approaches to stereo-matching adopt this universal approach when attempting to establish a single cost function for appraising the similarity of potentially matching correspondences. In this chapter, it is proposed that a more robust approach will be to adaptively establish matching conditions based on specific properties of the hand (e.g. skin tone etc.). Last, that the most effective approach to stereo correspondence search is a coarse-to-fine one, which can implicitly manifest in a machine learning context.

Contributions: This chapter proposes a novel, data-driven Regressive Random Forest framework that learns the direct mapping between a stereo image pair and high-quality groundtruth depth measurement.

In so doing, it presents Conditional Regressive Random Forest (CRRF), an innovative combination of Regressive Random Forest and Conditional Random Fields to model this mapping. A major contribution of this research is the use of a machine learning framework to combine various stereo matching criteria (multiple cost functions and window sizes) with the aim of implicitly determining stereo correspondences.

Unlike conventional CRF methods that require iterative solutions, a closed form solution to CRRF inference is derived. Similar to the framework previously presented, the CRRF framework has much wider application, particularly to problems that can be posed using graph theory.

5.1 Overview of Conditional Regressive Random Forest (CRRF)

The method proposed in this chapter recovers a high-quality depth image from two stereoscopically acquired images of the hand. Figure 5.2 shows an overview of the approach. First, the reference stereo image is segmented into superpixels using Simple Linear Iterative Clustering (SLIC) [107]. For every superpixel that lies within the hand region, its stereo-matching cost with all potentially matching pixels along the epipolar line in the corresponding image is computed. Five different matching cost functions were applied simultaneously (these include: Sum of Absolute Difference (SAD), Sum of Squared Differences (SSD), Normalized Cross Correlation (NCC), Quantized Census (QC), and Zero-mean Sum of Absolute Differences (ZSAD)). The reader is referred to Section 2.1.1 and [108] for details on these cost functions. Each of these stereo-matching cost functions is applied under varying window sizes that are centered on the centroid of

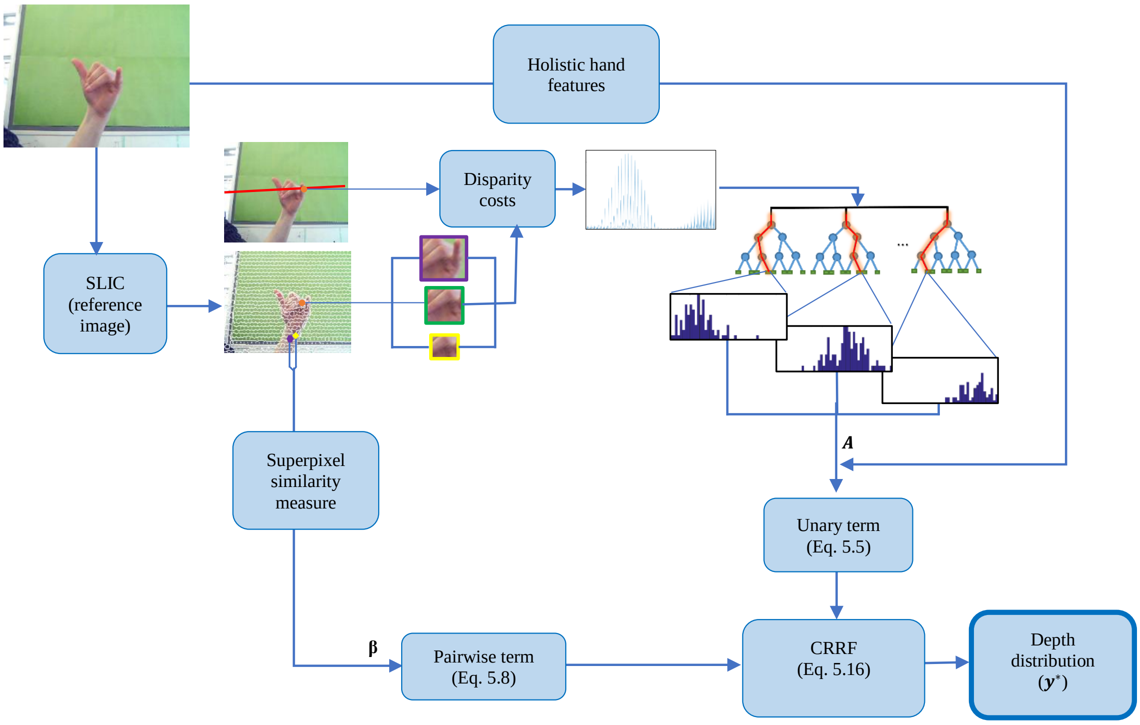


Figure 5.1 An illustration of the proposed approach. First, the reference stereo image is segmented into superpixels. Using different window sizes and cost functions, the disparity cost along the epipolar line in the corresponding image is computed. This cost is concatenated to generate a feature signal that is fed into a Regressive Random Forest. Posterior probability distributions from the trees are combined using the matrix, $\boldsymbol{A}$ to bias against the Holistic hand features of the hand. This yields the unary term of the CRRF model). The similarity measure between neighboring superpixels is multiplied with $\boldsymbol{\beta}$ to yield the pairwise term. The CRRF is solved in a closed form solution, y^*, that maximises Eq. 5.9.

the superpixel, and on the potentially matching pixels in the corresponding stereo pair. The matching cost values that are computed across all combinations of cost functions, window size and potentially matching pixel are concatenated to a single feature vector. Henceforth, this vector of features will be referred to as the *matching-cost feature vector*. Note that the proposed approach does not attempt to identify matching pixels explicitly; it simply computes the matching-cost feature vector (for each superpixel). In addition, features that relate to the hand in the scene are also extracted. These features primarily represent how far away the entire hand is from the camera, texture, and the color of the skin and will be referred as the *holistic hand feature vector.* The holistic hand feature consists of three main factors. These include the average intensity value of all hand region pixels; the aggregative shift of all hand pixels in the reference stereo camera compared to the other stereo camera; and lastly the ratio between the number of hand and non-hand region pixels is computed. This results in a six-dimensional holistic hand feature vector (3 color channels values, 2 vector shift values, and 1 ratio of pixels in the hand vs. non-hand regions). More detail is presented in Section 5.2.1.

A Regressive Random Forest (RRF) is trained to regress for the depth of a superpixel based solely on its matching-cost feature, however, each tree in the RRF is exposed to a subset of the training data based on its holistic hand feature. Finally, a CRF framework is used to combine the predictions from each tree in the RRF whilst constraining for smooth depth surface prediction. This combined framework is referred to as Conditional Regressive Random Forest (CRRF). The following sections delve into greater detail of approach.

5.1.1 Notation

For a given reference image, z, and its corresponding stereo image, z', a hand superpixel in z is denoted as $x_j \in \{x_1, ..., x_J\}$ and the centroid pixel of the superpixel as $\boldsymbol{v}_j$. For each $\boldsymbol{v}_j$, there exists a search space of W potentially matching pixels, $\boldsymbol{v}_{j,w} \in \{\boldsymbol{v}'_{j,1}, ..., \boldsymbol{v}'_{j,W}\}$ located in z'. The vector

$$\boldsymbol{c}_{k,g}(\boldsymbol{v}_j) = [f_{k,g}(\boldsymbol{v}_j, \boldsymbol{v}'_{j,1}), f_{k,g}(\boldsymbol{v}_j, \boldsymbol{v}'_{j,2}), ..., f_{k,g}(\boldsymbol{v}_j, \boldsymbol{v}'_{j,W})], \tag{5.1}$$

where $f_{k,g}$ is the resulting cost from using the k^{th} matching cost function, and g^{th} window size. $\boldsymbol{c}_{k,g}(\boldsymbol{v}_j)$ is concatenated for all combinations of k and g to get a single matching-cost feature vector. Hence for each superpixel, x_j, given that $k \in \{1, ..K\}$ and $g \in \{1, ..G\}$, the corresponding matching-cost feature will be $\boldsymbol{c}_j \in \mathbb{R}^N$ where $N = W * G * K$. Note that W, G and K are the number of pixels in the search space, the number of window sizes, and the number of matching cost functions respectively.

The groundtruth depth at the centroid pixel,$\boldsymbol{v}_j$, is d_j, the regression dataset is then defined as $\{(d_1, \boldsymbol{c}_1)^{(z)}, ..., (d_J, \boldsymbol{c}_J)^{(z)}\}$ for all Z stereo image pairs collected over different hand poses and subjects. The extracted feature is fed into the Random Forest based framework. The Random Forest framework is described in the next subsection.

5.1.2 Expert Random Forest

Decision trees are grown by recursively splitting and passing training data as described in Section 4.2.1. Here the matching-cost feature is used to determine the split. The intuition is that the trees implicitly learn how to adaptively select the size of window and type of cost function based on different tree split levels. This is analogous to adaptively determining the size of the window and type of cost function to use at different stages of a coarse-to-fine approach to search for pixel correspondence. The entropy decreases moving through each tree from the root node to the leaf nodes. Experimental results will show that the entropy is related to the coarse-to-fine selection of features.

Expert Trees: As previously stated, holistic hand features (features that describe the entire hand), are additionally computed. This step is motivated by the significant effect that features like skin color and the overall distance of the hand have on the matching-cost features. Consequently, establishing a stereo-matching criterion (i.e., matching cost, window size, etc.) that works effectively across different skin tones and hand depth levels is a difficult task. To this end, all the stereo image pairs are clustered into classes based on their holistic hand features. Each tree in the RRF is trained by bagging from only one of the classes, making it an expert at regressing the depth for that class. Thus, a particular tree may be expert at predicting the depth of superpixels in a darker-toned hand that is closer to the camera, whilst another may specialize in lighter-toned hands that are further away. See Section 5.2.1 for more detail on holistic hand features. When predicting the depth of an unseen stereo pair with a holistic hand feature, the CRF framework, discussed in the next subsection, ensures that more emphasis is placed on prediction from expert trees with similar holistic hand features than to others. Now the Random Forest framework is established, the next subsection discusses how the CRRF framework selectively introduces a bias for each class of tree using the holistic hand features whilst accounting for continuous depth estimate.

5.1.3 CRRF Framework

Consider a new stereo image pair, with a holistic hand feature vector, $\boldsymbol{h}$, whose superpixels' depths are to be predicted using the trained RRF. For a single superpixel, x_j, each RRF tree, t, produces a posterior probability distribution, $p_t(d_j|\boldsymbol{c}_j)$. This distribution is discretized by quantizing the depth values into D finite values. This yields a probability vector, $\boldsymbol{p}_{t,j} \in \mathbb{R}^D$ that is then consolidated across all the T trees

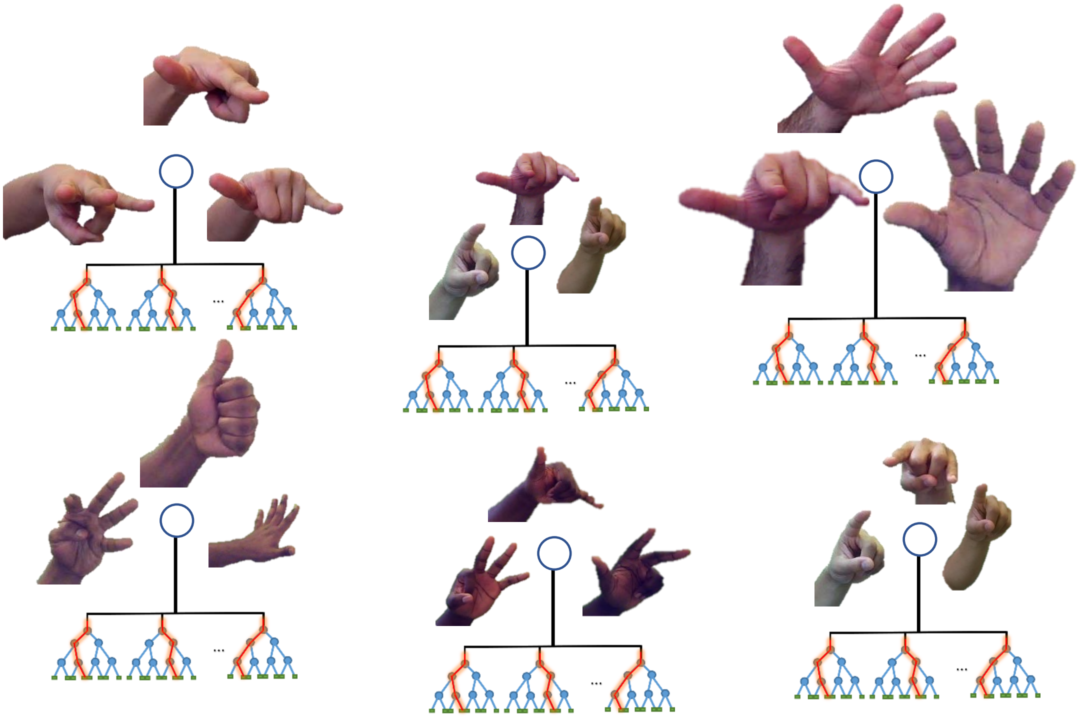


Figure 5.2 An illustration of expert trees and their training clustered data. First, the dataset is clustered based on their holistic hand features (that describes the skin tone, distance of hand from camera etc.). Each group of expert trees is trained only on a single cluster making them expert that that class of hand (e.g. a light-toned hand that is afar).

into $\boldsymbol{P}_j = [\boldsymbol{p}_{1,j}, \boldsymbol{p}_{2,j}, ..., \boldsymbol{p}_{T,j}] \in \mathbb{R}^{D \times T}$. The probability of d_j, given the reference stereo image and trained RRF, $Pr(d_j|\boldsymbol{P}_j, \boldsymbol{h})$, is modelled as a CRF. Conventionally a CRF formulates conditional probability as a product of potentials, that is

$$Pr(a|b) = \frac{1}{Z(b)} \prod_i \exp(\phi_i) = \frac{1}{Z(b)} \exp\left[\sum_i (\phi_i)\right], \tag{5.2}$$

where $Z(b)$ is the partitioning function, and ϕ_i are potentials. See Section 3.2.6 for more detail. Inspired by the work of Liu et al. in [109], the potentials in the proposed framework take the form of a unary E_U and a pairwise term E_P.

[109] presents a deep convolutional neural field model for estimating depths from a single image by integrating the capacity of deep CNN with a continuous CRF. Specifically, it proposes a framework which learns the unary and pairwise potentials of a continuous CRF in a unified deep CNN framework. The proposed method is applied to depth recovery of monocular capture of generic scenes. Here the integral of the partition function was analytically calculated, thus providing a solution for the log-likelihood optimization. In the framework presented in this chapter, conditional probability is approximated because of the intractable nature of $Z(b)$ (as it requires an integral over all combination of all possible states that the target and input variable could have)in the proposed framework,

$$\widetilde{Pr}(\boldsymbol{d}_j|\boldsymbol{P}_j, \boldsymbol{h}) = \exp\left[\sum_c (\phi_c)\right] = \exp[E_U + E_P], \tag{5.3}$$

where $\widetilde{Pr}$ denotes an unnormalized probability distribution. This approximation will suffice because the objective is to estimate the depth level with the maximum probability. Hence, the probability of the depth distribution function for all superpixels given $\boldsymbol{P}_j$ and the image's holistic hand feature, $\boldsymbol{h}$, is represented as the exponent of sums of both potentials. While the unary term aims in yielding a conditional probability distribution that maximizes the probability of the true depth level, the pairwise term encourages neighboring superpixels to have a similar posterior probability distribution.

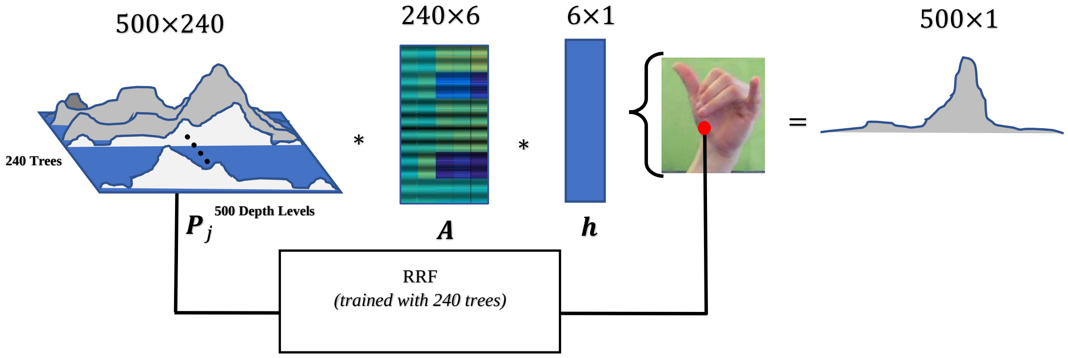


Figure 5.3 An illustration of the unary potential when the number of trees, $T = 240$, the number of depth levels, $D = 500$ and the number of holistic hand features, $H = 6$. This illustrates how $\boldsymbol{A}$ weights the posterior probability, $\boldsymbol{P}_j$, from the trees using $\boldsymbol{h}$ to give a probability distribution of a single superpixel. This becomes the unary term in the CRRF.

Unary Potential: The unary term predicts the depth level of a superpixel based on its posterior distribution from the RRF trees and the holistic hand feature. To this end a unary weighting matrix, $\boldsymbol{A} \in \mathbb{R}^{T \times H}$, is introduced, which weights the posterior from each tree based on $\boldsymbol{h} \in \mathbb{R}^{H}$. This is important because expert trees are trained, as opposed to randomly bagged trees. The $\boldsymbol{A}$ matrix provides weights to trees depending on the holistic hand feature. Hence it places varied emphasis on the predictions from different trees.

Taking inspiration of the Bhattacharyya metric [110], E_U is formulated as an affinity measure between true depth probability, $\widehat{\boldsymbol{p}}_j^T$, and the predicted probability, $\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}$, as in,

$$E_U = \frac{1}{J}\sum_{j=1}^{J}\left[\frac{\widehat{\boldsymbol{p}}_j^T \boldsymbol{P}_j \boldsymbol{A}\boldsymbol{h}}{\boldsymbol{i}^T \boldsymbol{A}\boldsymbol{h}}\right]. \tag{5.4}$$

This is accumulated across all superpixels in the reference stereo image. The denominator in Eq. 5.4 ensures that $\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}$ remains normalized. The surface plot in Figure 5.4 shows how the different entries of $\boldsymbol{A}$ vary relatively. Figures 5.3 and 5.4 give an illustration of the weighting ability of $\boldsymbol{A}$. The peaks indicate a strong relationship between entries of $\boldsymbol{h}$ and the tree index. Studying both figures, consider a hypothetical example where $\boldsymbol{h} = [0, 0, 0, 1, 1, 1]^T$. In this case, the holistic hand feature vector will weight the prediction from the 240 trees based on the last three columns of $\boldsymbol{A}$, thereby giving less weighting to trees 40 to 80 and trees 160 to 200.

Let $\widehat{\boldsymbol{y}} = [\widehat{\boldsymbol{p}}_1^T, \widehat{\boldsymbol{p}}_2^T, ..., \widehat{\boldsymbol{p}}_j^T] \in \mathbb{R}^{(D*J)}$ be a vector resulting from the concatenation of the actual probability distribution of all hand region superpixels and let $Y = [\boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_J]^T \in \mathbb{R}^{(D*J)\times T}$ be the matrix whose row vectors are the concatenation of the predicted probability distribution from each tree. Then the unary potential in Eq. 5.4 can be rewritten for all superpixels in a single stereo image, z, in matrix form as follows:

$$E_U = \frac{1}{J\boldsymbol{i}^T\boldsymbol{A}\boldsymbol{h}}\widehat{\boldsymbol{y}}^T\boldsymbol{A}\boldsymbol{h}. \tag{5.5}$$

The larger E_U becomes, the more similar the consolidated predicted probability, $\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}$, is to the true depth probability, $\widehat{\boldsymbol{p}}_j^T$.

Pairwise Potential: The pairwise potential enforces the constraint that adjacent superpixels often possess similar depth and hence similar probability distributions. This is based on the smooth nature of the depth of the hand surface. Similar to [109], a

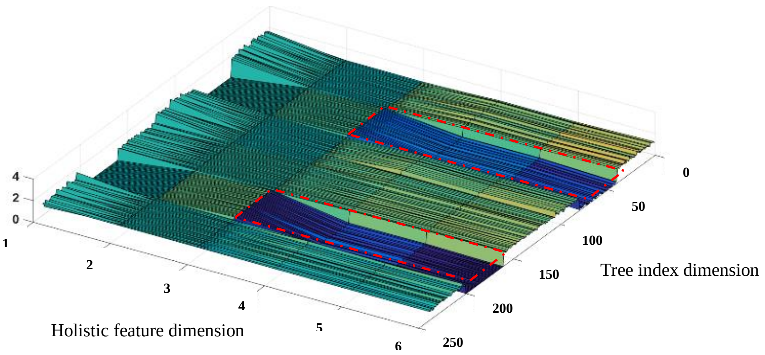


Figure 5.4 A surface plot of the matrix $\boldsymbol{A}$ (see Figure 5.3), used to weigh the expert trees based on the holistic hand feature. A higher value indicates more weight. Consider a hypothetical holistic hand feature vector, $[0, 0, 0, 1, 1, 1]$, which, when post-multiplied with $\boldsymbol{A}$ will give less weighting to trees 40 to 80 and 160 to 200 based on their lower values (bluer colours), highlighted with red boxes.

visual similarity measure between neighborhood superpixels is established to apply an adaptive depth similarity constraint. Specifically, neighboring superpixels that appear dissimilar in terms of color, texture, and size will have a weaker pairwise potential encouraging similar predicted depth. This is particularly intuitive in a self-occluded scenario. The discontinuity in texture resulting from a finger occluding the palm, for example, will indicate that a lower smoothness constraint is placed on neighboring superpixels that exist on the edge of the finger and the palm.

To achieve this behaviour, a similarity vector, $s_{j,k} = \left[s^1_{j,k}, ..., s^Q_{j,k}\right]$, and a pairwise weighting, $\boldsymbol{\beta} \in \mathbb{R}^Q$, are introduced. For a pair of neighbouring superpixels, x_j and x_k, Q superpixel similarity measures are computed between them (more details on the superpixel similarity measures are presented in Section 5.2.1). Pairwise potential is specified as:

$$E_P = \frac{1}{|U|} \sum_{(j,k)\in U} \boldsymbol{\beta}^T \boldsymbol{s}_{j,k} \widehat{\boldsymbol{p}}_k^T \widehat{\boldsymbol{p}}_j \tag{5.6}$$

where U is a set of all possible pairs of neighbouring hand superpixels. Subsequently, the pairwise potential is a measure of the affinity of the probability of all pairs of neighbouring superpixels, and $\boldsymbol{\beta}^T \boldsymbol{s}_{j,k}$ determines the contribution of each pair of superpixels to this measure.

Let $\boldsymbol{B} \in \mathbb{R}^{J\times J}$ be a matrix such that, its elements are given by

$$\boldsymbol{B}_{j,k} = \boldsymbol{\beta}^T \boldsymbol{s}_{j,k} \boldsymbol{I}, \tag{5.7}$$

and zeros everywhere else. $\boldsymbol{I}$ is a $\boldsymbol{D} \times \boldsymbol{D}$ identity matrix. With this matrix, the pairwise potential in Eq. 5.6 can be represented in matrix form as:

$$E_P = \frac{1}{|U|} \widehat{\boldsymbol{y}}^T \boldsymbol{B} \widehat{\boldsymbol{y}}. \tag{5.8}$$

A resulting depth image with high level of smoothness will yield a large pairwise potential, E_P.

Complete CRRF: At this stage, both potentials, unary and pairwise, have been established and the higher they are, the smoother and the more likely the predicted depth becomes (based on its probability). Eqs. 5.3, 5.4 and 5.6 are combined to result in

$$\widetilde{Pr}(\boldsymbol{d}_j|\boldsymbol{P}_j,\boldsymbol{h}) = \exp\left[\frac{1}{J}\sum_{j=1}^{J}\left[\frac{\widehat{\boldsymbol{p}}_j^T\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}}{\boldsymbol{i}^T\boldsymbol{A}\boldsymbol{h}}\right] + \frac{1}{|U|}\sum_{(j,k)\in U}\boldsymbol{\beta}^T\boldsymbol{s}_{j,k}\widehat{\boldsymbol{p}}_k^T\widehat{\boldsymbol{p}}_j\right], \tag{5.9}$$

for a single stereo image pair. In this unified framework, the aim is to maximize Eq. 5.9 based on $\boldsymbol{A}$ and $\boldsymbol{\beta}$. For all stereo images in the training set, z, the framework attempts to maximize $\sum_z \log\widetilde{Pr}(\boldsymbol{y}^{(z)}|\boldsymbol{P}^{(z)})$. Formally,

$$\max_{\boldsymbol{A}\geq\boldsymbol{0},\boldsymbol{\beta}}\sum_{z=1}^{Z}\log\widetilde{Pr}(\boldsymbol{y}^{(z)}|\boldsymbol{P}^{(z)}) + \lambda(1-\boldsymbol{\beta}^T\boldsymbol{\beta}) \tag{5.10}$$

where λ is the decay weight on the constraint with $\boldsymbol{\beta}$ maintaining a unit length and

$$\log\widetilde{Pr}(\boldsymbol{d}_j|\boldsymbol{P}_j,\boldsymbol{h}) = \frac{1}{J}\sum_{j=1}^{J}\left[\frac{\widehat{\boldsymbol{p}}_j^T\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}}{\boldsymbol{i}^T\boldsymbol{A}\boldsymbol{h}}\right] + \frac{1}{|U|}\sum_{(j,k)\in U}\boldsymbol{\beta}^T\boldsymbol{s}_{j,k}\widehat{\boldsymbol{p}}_k^T\widehat{\boldsymbol{p}}_j. \tag{5.11}$$

The monotonic nature of log functions implies that Eq. 5.11 increases as $\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}$ and $\widehat{\boldsymbol{p}}_J^T$ becomes more similar and the resulting depth becomes more smooth. During optimization, it is ensured that all the entries of $\boldsymbol{A}$ are positive, so that $\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h}$ represents a probability. With the aim of solving for Eq. 5.10, stochastic gradient ascent is applied using the partial derivative of Eq. 5.11 with respect to $\boldsymbol{A}$ and $\boldsymbol{\beta}$:

$$\frac{\partial\{\log\widetilde{Pr}(\boldsymbol{y}|\boldsymbol{P},\boldsymbol{h})\}}{\partial\boldsymbol{A}} = \frac{1}{J}\sum_{j=1}^{J}\frac{\boldsymbol{P}_j^T\widehat{\boldsymbol{p}}_j\boldsymbol{h}^T(\boldsymbol{i}^T\boldsymbol{A}\boldsymbol{h}) - (\widehat{\boldsymbol{p}}_j^T\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{h})\boldsymbol{i}\boldsymbol{h}^T}{[\boldsymbol{i}^T\boldsymbol{A}\boldsymbol{h}]^2} \tag{5.12}$$

and

$$\frac{\partial\{\log\widetilde{Pr}(\boldsymbol{y}|\boldsymbol{P},\boldsymbol{h})\}}{\partial\boldsymbol{\beta}} = \frac{1}{|U|}\sum_{(j,k)\in U}\boldsymbol{s}_{j,k}^T\widehat{\boldsymbol{p}}_j\widehat{\boldsymbol{p}}_k^T. \tag{5.13}$$

$\boldsymbol{A}$ and $\boldsymbol{\beta}$ are randomly initialized, and iteratively updated accordingly. See Section 5.2.3 for details.

Prediction: Having established $\boldsymbol{A}$ and $\boldsymbol{\beta}$, predicting the posterior probability for new stereo pairs involves solving the Maximum a Posteriori inference on Eq. 5.9. To achieve this, the matrix representations of E_P and E_U are used in Eq. 5.5 and Eq. 5.8

resulting in

$$\widetilde{Pr}(\boldsymbol{d}_j|\boldsymbol{P}_j, \boldsymbol{h}) = \exp\left[\frac{1}{|U|}\boldsymbol{y}^T\boldsymbol{B}\boldsymbol{y} + \frac{1}{N}\boldsymbol{y}^T\boldsymbol{Y}\boldsymbol{A}\boldsymbol{h}\right], \tag{5.14}$$

The aim is to determine $\boldsymbol{y}$ that maximizes $\widetilde{Pr}(\boldsymbol{y}|x)$ for a pre-computed $\boldsymbol{A}$ and $\boldsymbol{\beta}$ pair.

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} \widetilde{Pr}(\boldsymbol{y}|\boldsymbol{P}_j, \boldsymbol{h}) = \arg\max_{\boldsymbol{y}} \frac{1}{|U|}\boldsymbol{y}^T\boldsymbol{B}\boldsymbol{y} + \frac{1}{N}\boldsymbol{y}^T\boldsymbol{Y}\boldsymbol{A}\boldsymbol{h} \tag{5.15}$$

This is easily derived in closed form by solving for the zeros of the second derivative. Formally,

$$\boldsymbol{y}^* = \frac{|U|}{N}\boldsymbol{B}^{-1}\boldsymbol{Y}\boldsymbol{A}\boldsymbol{h}. \tag{5.16}$$

$\boldsymbol{y}^*$ represents the concatenated predicted depth probability for all superpixels in an image. The predicted depth level for a superpixel is the depth level with the maximum depth probability.

5.2 Implementation Details

When mapping the matching-cost features to groundtruth depth, it was important to establish a database of strong registration between the pairs of data. To this end the Dataset B introduced in Section 3.3.2 was used. This allows $\{(d_1, \boldsymbol{c}_1)^{(z)}, ..., (d_J, \boldsymbol{c}_J)^{(z)}\}$ to be established for all captured instance of stereo pairs, z. Dataset A was not used to avoid bias of learner to fronto-parallel poses.

5.2.1 Extracted Features

Matching-cost Features, $\boldsymbol{c}_j$: as discussed above, the implementation used five matching cost functions. These cost measures were chosen because of their prominence, computation cost, and simplicity. Of course, more complex types and combinations of matching costs could be used. Each of the cost functions was applied under three window sizes: [7×7], [11×11], and [15×15]. All combinations of these window sizes and matching costs were used to compare each centroid point in the reference stereo image to 50 potentially matching pixels (selected based on proximity to $\boldsymbol{v}_j$) that lie on the epipolar line in the corresponding stereo pair. This resulted in a 750-dimensional matching-cost feature vector being used to regress for the depth at each superpixel.

Holistic Hand Features, $\boldsymbol{h}$: For each captured instance of stereo pairs three main factors are used to describe the scene. First, the average intensity value of all hand region pixels across all three color channels is considered. This quantifies the skin tone. Second, the aggregative shift of all hand pixels in the reference stereo camera compared to the other stereo camera is computed. This quantifies how far away the hand is from the camera, representing the difference in the average pixel's position for hand region pixels in both cameras. Last, the ratio between the number of hand and non-hand region pixels is computed. This quantifies the size of the hand (if considered relatively to the aggregative shift). Note that the implementation of this technique is not limited to these three factors, the only constraint is that all entries of $\boldsymbol{h}$ must be positive values.

Superpixel Similarity Measure, $\boldsymbol{s}_{j,k}$: To quantify similarities of two neighboring superpixels four measures are used. The first measure is the difference in the average LAB color of both superpixels. The second is the difference in the Local Binary Pattern [111]. The third measure is the difference in the standard deviation of pixels' values in LAB color. Finally, the summed difference in histogram is examined. In each of these cases, the exponent of the negative Euclidean norm is applied to the resulting difference. E.g. the first entry (LAB difference) $\boldsymbol{s}_{j,k} = e^{-||\boldsymbol{s}_j^{LAB} - \boldsymbol{s}_k^{LAB}||}$, where $\boldsymbol{s}_j^{LAB}$ is the average LAB value for superpixel x_j. This yields a similarity measure vector with a length of four, or $Q = 4$. These features were chosen because of there success in discriminating neighbourhood affinity has illustrated in [109].

5.2.2 Random Forest

Recall that Dataset B (from Section 3.3.2) contains 500 instances of hand poses at different distances, from 12 participants (6,000 stereo pairs in total) of different skin

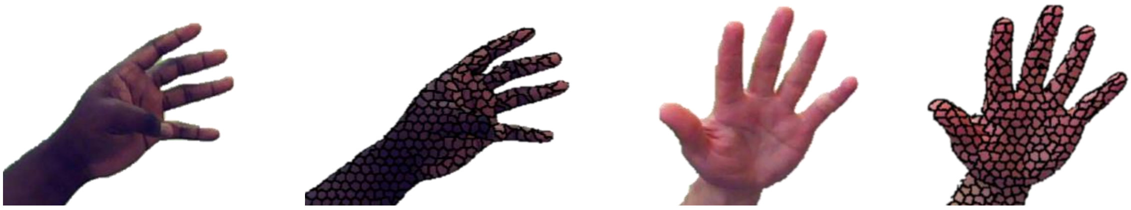


Figure 5.5 Examples of SLIC segmentation applied to hand region. Two original hand region images (1^{st} and 3^{rd}) and their corresponding SLIC-based segmentations (2^{nd} and 4^{th}).

tone, hand size, and gender. Data from four participants were reserved for testing, and the remaining data (from the other eight participants) was used for training. SLIC segmentation was applied to all reference stereo images, producing approximately 3,000 superpixels per image. Note that only a fraction of these 3,000 superpixels is hand region superpixels. The number of hand superpixels (ranging approximately from 200 to 500 per image capture) depends on the distance between the hand and the camera. In total, roughly 2.5 million superpixels were used in training and evaluating the algorithm. The depth value posterior distribution of the RRF was quantized into 500 bins, i.e. $D = 500$ (see Figure 5.5). The depth bin of 500 as it achieves a good balance between the precision of depth prediction and the size of matrix $boldsymbolB$. The depth range of the hand poses in the entire dataset generally ranged from $500mm$ to $1800mm$. Hence, the RRF can predict to a resolution of $(1800mm - 500mm)/500$ bins $= 2.6mm$.

With the focus on the training dataset (from the eight participants), first, all stereo pairs were clustered into six clusters based on the holistic hand feature (using k-means). The training data was divided into two sets (seven participants to one participant). The RRF was trained on the first set (containing data from seven participants) and then the second set (containing data from the remaining participant) was propagated from the trained RRF to acquire the posterior probability matrix, $\boldsymbol{Y}_Z$. This procedure was carried out iteratively for all permutations of seven training and one testing participant(s) in a cross-validation fashion, yielding a set of posterior probabilities $\{\boldsymbol{Y}_1^{(s)}, ..., \boldsymbol{Y}_Z^{(s)}\}$ of stereo images for training participant, s. Note that all $\boldsymbol{Y}_z^{(s)}$ estimations result from testing stereo images of training participant s on a RRF trained on images from all the other seven participants. All $\boldsymbol{Y}_z^{(s)}$ and $\boldsymbol{h}_z^{(s)}$ are subsequently used in the CRRF framework to estimate $\boldsymbol{A}$ and $\boldsymbol{\beta}$. The RRFs were trained in parallel on a cluster with MATLAB using two nodes, each with 20 processors. Each training round (i.e. to train for each posterior $\boldsymbol{Y}_z^{(s)}$) takes approximately 3 - 4 hours. Since eight rounds were needed, training took roughly one day. At test time, based on the MATLAB implementation, the SLIC algorithm runs in 38.23 seconds on average to segment a single reference image. The extraction of the holistic hand and stereo-matching features takes 36.34 seconds on average for each stereo pair. Finally, the propagation of all superpixels and combining the posteriors using $\boldsymbol{\beta}$ executes typically in 185 seconds. Hence testing for the depth a frame of stereo images on the cluster will typically take 260 seconds. Note that the runtime could be considerably reduced in future work by recoding the method in C++ and using GPU techniques.

5.2.3 Stochastic Gradient Ascent

$\boldsymbol{A}$ and $\boldsymbol{\beta}$ are learned separately by first randomly initializing, with all elements of $\boldsymbol{A}$ being positive. First $\boldsymbol{A}$ is trained for and then $\boldsymbol{\beta}$ is learnt under a fixed $\boldsymbol{A}$. In both cases, the learning rate was initialized at 12,000. Training was carried out on 100 epochs, reducing the learning rate by 10% every 10 epochs. The decay weight, λ, was set as 0.05. For greater clarity, the entire framework is summarised, identifying and outlining key features and how they relate in Table 5.1.

5.3 Experiments and Results

The approach was validated experimentally, presenting both qualitative (Figure 5.7) and quantitative (Table 5.2) results. Three main comparisons were made, these were prediction solely using RF (with only matching-cost features and with a combination of matching-cost and holistic features); using RF with the unary term framework; as well as the ELNF technique presented in the previous chapter. The results were quantitatively

Components	**Implication**
Matching-cost features (per superpixel)	This feature vector describes how similar/dissimilar the centroid of the superpixel is to all pixels along the epipolar line on the corresponding stereo image. This is potentially determined by the disparity at that centroid pixel.
Superpixel Similarity measure	This is a vector of metrics that conveys how similar or dissimilar two neighboring superpixels are.
Holistic Hand features	This feature vector describes the general shift, tone, and size of the hand.
Expert Trees	RRF are conventionally built, however, each tree is trained on a dataset of hands captures of a particular class, based on its Holistic hand feature.
Unary Term	During a superpixel depth prediction, the unary term facilitates the bias to predictions from expert trees that were trained from a dataset of similar Holistic hand features.
Pairwise Term	The pairwise term adds the constraint that superpixels depth predictions yield a continuous surface in a neighborhood, i.e. neighboring superpixels (particularly those with high Superpixel Similarity measure) will tend to have similar depth predictions.
CRRF Formulation	The CRRF formulation yields a closed form solution to superpixel depth prediction that combines the unary and pairwise terms.

Table 5.1 A Brief outline of key components of the proposed framework. This includes Matching-cost feature, Holistic Hand feature, Superpixel Similarity measure, Expert trees, Unary Term, Pairwise Term, and the CRRF formulation.

appraised for accuracy by computing the percentage of correctly predicted depth both at superpixel and pixel levels, $\frac{\sum_{p \in N} \mathbb{I}[|d_p^{GT} - d_p| < t]}{N}$, where d_p^{GT} and d_p are the groundtruth and the predicted depth at superpixel (or pixel) p; $\mathbb{I}[]$ is a function that returns 1 for true input and 0 otherwise; and N is the number of hand region pixels/superpixels. The average relative error, $\frac{1}{N} \sum_{p \in N} \frac{|d_p^{GT} - d_p|}{d_p^{GT}}$, was computed to quantitatively evaluate the performance of the test. The following subsections will review the results in Table 5.2 and Figure 5.7 in more detail.

5.3.1 Stereo-matching Comparison

To validate the machine learning approach, depth recovery (through disparity) from stereo pairs in Dataset B using a prominent stereo-matching technique, SGM was performed. At the time of writing, this was the 9^{th} best performing published stereo-matching technique on the Middlebury stereo evaluation chart [112]. SGM was chosen as it is the highest performing technique for which a MATLAB implementation is readily available.

The same calibration information used in establishing the epipolar line of Dataset B was used to rectify the stereo capture of hand poses. Then the rectified stereo pair was fed into the MATLAB implementation of SGM for stereo-matching. Stereo baseline and focal length resolved from stereo calibration are combined with the SGM generated disparity to yield the actual distance. Performance error is computed based on hand pixel regions. This is shown in Figure 5.7 and Table 5.2 (last row). This is an interesting comparison as SGM also applies global optimization. Nonetheless, its poor performance is apparent from Table 5.2. It provides the least accuracy and the most

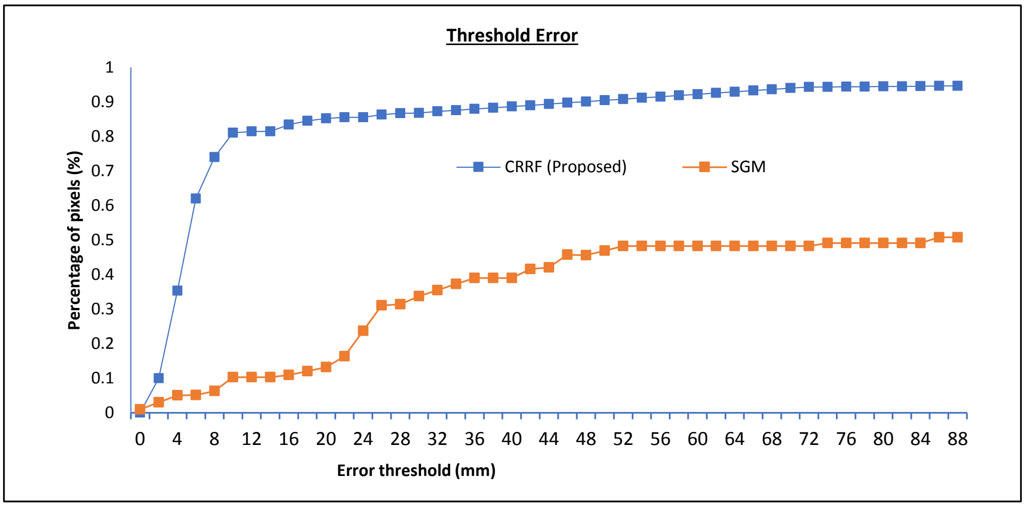


Figure 5.6 This graph compares the pixel level accuracy of CRRF to SGM as the threshold value varies. The superiority of CRRF is most apparent at higher depth error threshold.

error in comparison to the rest of the machine learning techniques. The hypothesis here is that this is due to the untextured nature of the hand as well as radiometric differences present in the stereo pair. The SGM technique attempts to universally appraise pixel correspondence by applying a pre-established matching criterion. The untextured nature of the hand and radiometric inconsistencies, in conjunction with the varying skin colors and hand sizes, makes this task hard. This result emphasizes the significance of the proposed approach in that a conventional stereo-matching approach (even one as robust as SGM) performs poorly for skin regions. Further investigation on the performance of the two techniques was performed using pixel level accuracy with a varying threshold. The graphical comparison is presented in Figure 5.6. Again, the superiority of CRRF is demonstrated. A significant result is that a high percentage of depth predictions made using the proposed approach are accurate in comparison to SGM. However, as the error threshold gets closer to $8mm$ the percentage drops abruptly. To put this into context, the smallest finger on a hand is typically $10mm$ in width. Hence, at least 81% of the structure of the fingers are mostly discernable. This contrasts with 10.3% in the case of SGM.

5.3.2 Baseline Comparison

Four baseline comparisons were made. The first was predicting depth solely from the matching-cost feature, using conventional RRF. The results (Table 5.2) validate the hypothesis that applying a machine learning approach to learning the stereo-matching criteria for determining stereo correspondence is a more effective approach. Using a set of simple stereo-matching criteria and stochastically determining which to use at different tree depths has resulted in almost a 272.7% increase (from 0.132 to 0.492) in pixel level accuracy.

Secondly, the matching-cost feature was augmented by concatenating it with the holistic hand features whilst still regressing with a conventional RRF model. The aim was to specifically investigate the impact of using “expert trees”. From Table 5.2 one can see a notable improvement in the prediction resulting from adding the holistic

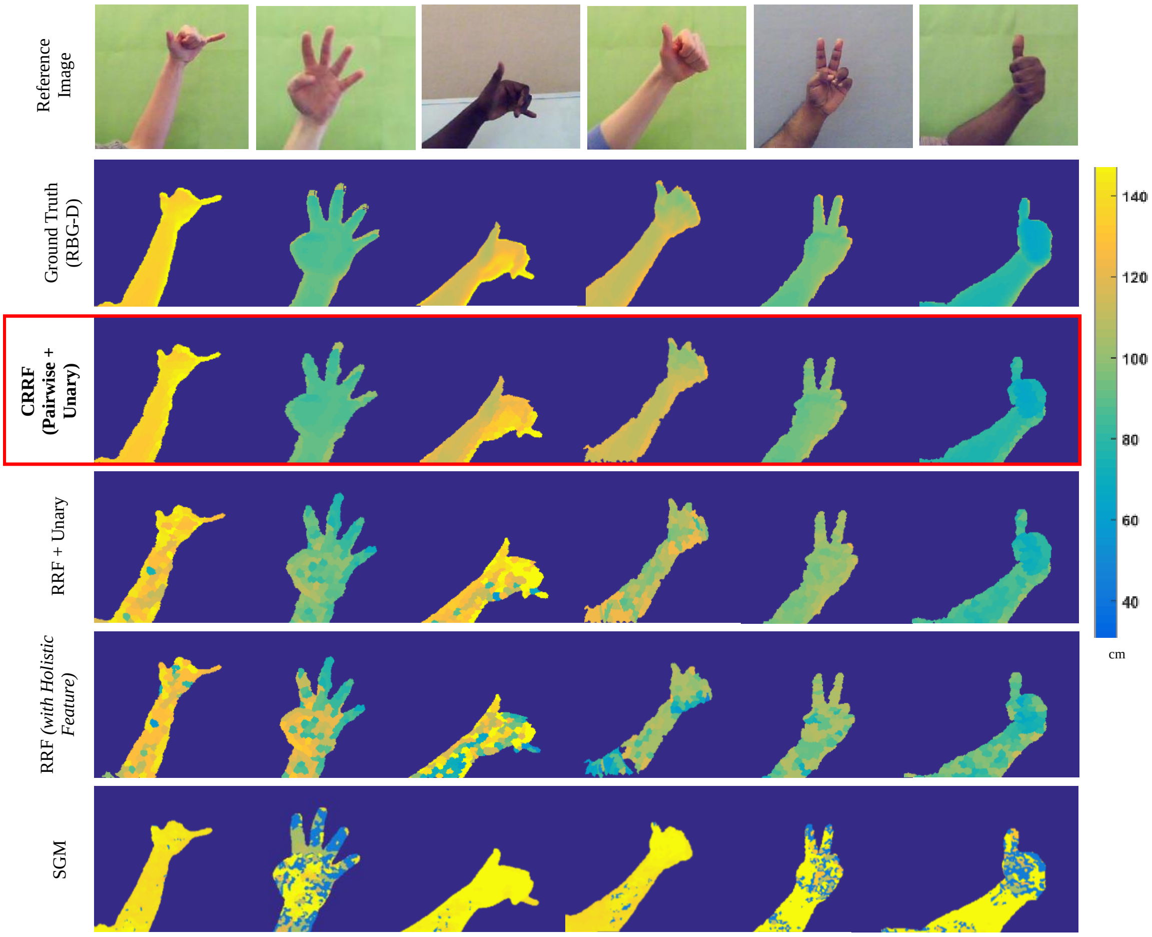


Figure 5.7 Qualitative Results using real captured poses. The reference image of the stereo pair is shown in the 1^{st} row and the corresponding groundtruth depth is presented in the 2^{nd} row. The results from the proposed technique are presented in the 3^{rd} row. Results from solely using the unary term with RF are in the 4^{th} row, while recovered depths from RF are presented in the 5^{th} row. The quality of the recovered depth as a result of CRRF is apparent.

feature, yielding greater accuracy (0.492 to 0.689) and less relative error (0.500 to 0.353) in both superpixel level and pixel level. However, a much greater increase in accuracy results from using the holistic feature to learn expert trees as opposed to just concatenating it with the stereo-matching feature. This yielded a 50.2% increase in accuracy on average in comparison to the 29.1% increase in accuracy provided by solely concatenating the holistic features. The last baseline comparison was to investigate the significance of the pairwise term. Recall that the contribution of the pairwise term is to

Methods	**Superpixel Accuracy**		**Pixel Accuracy**		**Ave. Relative Err.**	
	t=10mm	**t=20mm**	**t=10mm**	**t=20mm**	**per Superpixel**	**per Pixel**
SGM	-	-	0.103	0.132	-	0.772
ELNF	-	-	0.455	0.515	-	0.534
RRF	0.599	0.610	0.423	0.492	0.503	0.500
RF (with Holistic Feature)	0.686	0.757	0.610	0.689	0.358	0.353
RF + Unary	0.835	0.885	0.684	0.788	0.229	0.231
CRRF (Pairwise + Unary)	**0.911**	**0.911**	**0.811**	**0.852**	**0.181**	**0.190**

Table 5.2 Quantitative comparison of the proposed technique (CRRF + Pairwise + Unary) against the ELNF (proposed in the previous chapter), conventional RRF, and different variants of the proposed technique.

add a smoothing constraint on the depth prediction. This is presented in the qualitative results. The predicted depth is clearly smoother and hence a better representation of the surface of the hand. The quantitative result from Table 5.2 also conveys the superiority of the prediction made when the pairwise term is applied. Interestingly, the pixel level accuracy is almost as strong as the superpixel level accuracy when the pairwise term is applied. This is again due to the smoothing effect. Although the superiority of the proposed approach against the baseline comparison is evident, there still exists some failure cases (see Figure 5.8). It can be observed that approach fails in scenarios were the hand positioned in extreme distance (i.e. too far) away from the camera.

5.3.3 Comparison with ELNF

Comparison with the ELNF framework presented in the previous chapter was made. ELNF also applies a Regressive Random Forest to estimate image depth. However, a single similarity measure (Quantized Census) was used to compute depth image, and no pairwise term is modeled in the regression that maps a disparity image to a depth image. As the results in Table 5.2 show, the proposed method, even without the pairwise term, outperforms ELNF. This improved performance in the proposed method

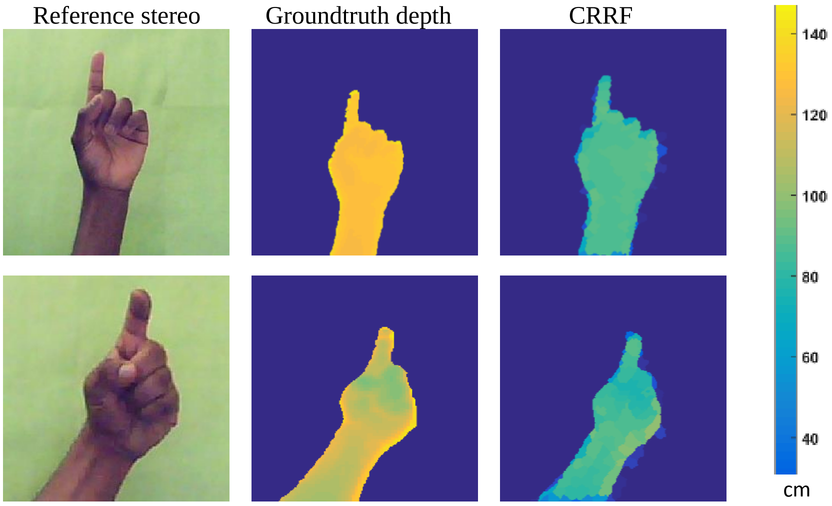


Figure 5.8 Two examples of scenarios where the CRRF produces weak results. The Left image is the reference stereo capture whilst the middle image shows the groundtruth depth. The predicted depth image is shown on the right.

is attributed to the features used. Unlike ELNF, which uses a single similarity measure, the proposed method learns the features that best regress the depth using multiple similarity measures, disparity shifts, and window sizes in a concatenated feature vector. Also unlike ELNF, which uses disparity as an intermediate representation, the proposed method maps directly from the stereo pair to depth. Additionally, the CRRF framework's regression is more sophisticated in that it conditionally learns expert trees, which are combined using holistic hand features. Finally, the pairwise term in the proposed model provides additional smoothing constraints that yield superior performance.

5.3.4 Evaluating Performance vs Depth Range

The performance of the CRRF technique at different depth levels is investigated here. To this end, the average error for pixels of a particular depth range was experimented with. The results are presented in Figure 5.9. Figure 5.9a compares the performance of the CRRF technique to SGM. It can be observed that in both cases, depth prediction for pixels closer to the camera is relatively poor (higher error). The prediction performance

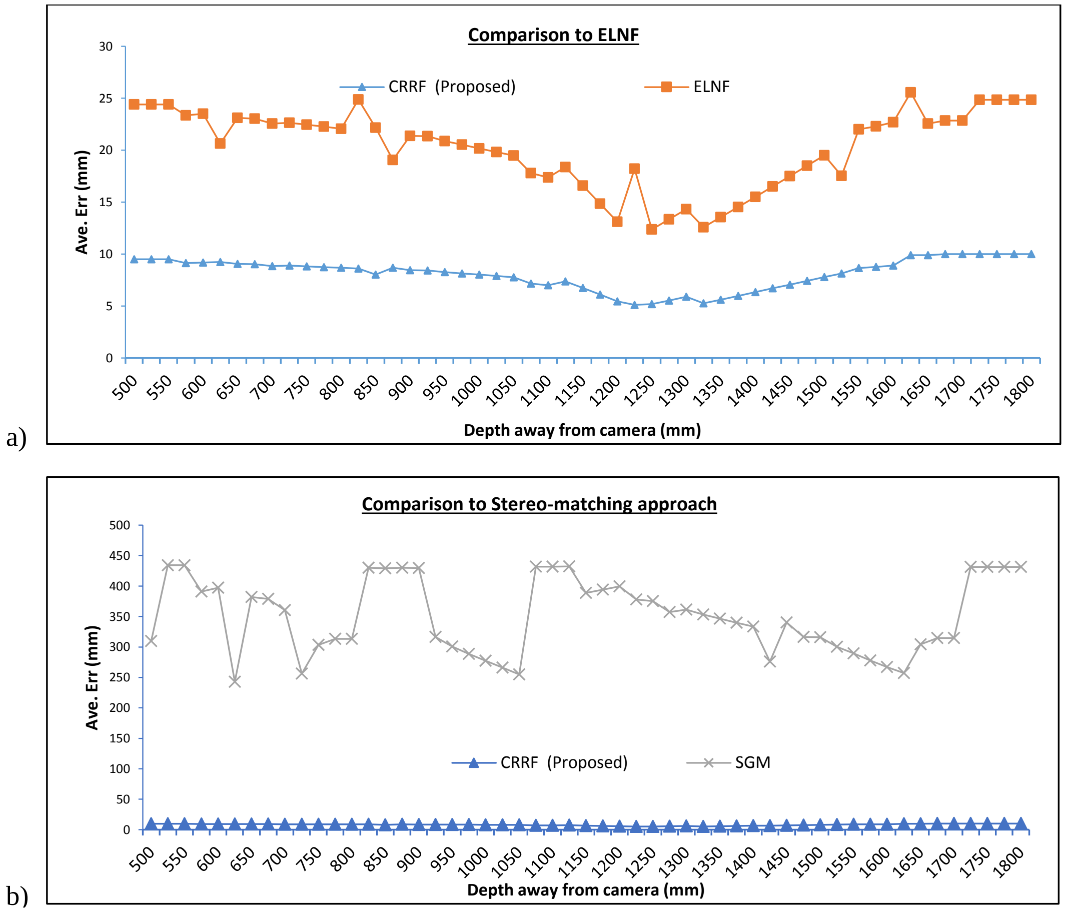


Figure 5.9 An illustration of performance over depth levels. The first graph (a) compares the performance of CRRF to the ELNF approach presented in the previous chapter whilst the second graph (b) compares CRRF to using SGM respectively, over different superpixel depth levels.

increases for pixels that exist closer to the middle of the depth range. A dip in performance appears again for the remaining pixels. This trend in performance (that is consistent with the machine learning based approach) is not shared by the performance of SGM depth recovery (Figure 5.9b). The variation in the performance of SGM is less systematic. A possible reason for this is the quality of groundtruth depth in these regions. The fact that the groundtruth (RGBD recovered depth) itself has a low-quality depth measurement at lower and at higher depth could yield poor depth prediction at these depth values. Since at the extreme depths (too close and too far from the

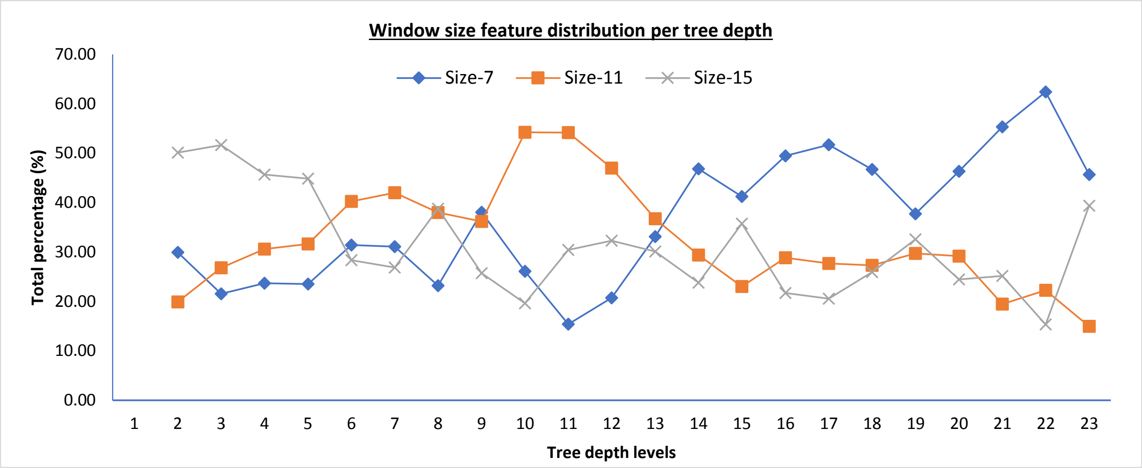


Figure 5.10 Graphical illustration of the distribution of feature selection at different depth levels of the regression forest. Specifically, it shows the total percentage of evaluated **7 × 7**, **11 × 11**, and **15 × 15** window features at different tree depth levels. This corroborates the conjecture that at shallow tree levels the trees are biased to a particular matching criterion. In this case, the larger window sized features (**15 × 15**) are evaluated more at shadow tree depth and vice versa.

camera), the model is learning from a lower confidence (and potentially less consistent) dataset.

5.3.5 Evaluating the Coarse-to-fine Conjecture

As stated in the Section 5.1, the approach was motivated by aiming to implement a coarse-to-fine framework in a machine learning context. This section investigates to what extent the RRF exhibits this coarse-to-fine feature. To do so, during training (of the RRF) all superpixels entering all nodes at each tree depth level were collected and the percentage of superpixels that were evaluated at a particular feature type calculated, keeping in mind that each superpixel that propagates through the RRF possesses a matching-cost feature vector where each of the elements corresponds to a particular window size and matching cost function. Hence, for a superpixel entering a node, the feature position that was evaluated is examined (to determine the split) and tallied. The same applies to matching cost and window size to which the feature corresponds.

The results are presented in Table 5.13a to 5.13c. Note that the percentile is computed across each depth level. Looking at the table, it can be noted that the RRF prefers different types of features at different depth levels. For instance, 7 × 7 and 11 × 11 window sized SAD features are less evaluated at shallow tree depth (depth levels 4 to 10). The same applies to 15 × 15 window sized Quantized Census features at deeper tree levels. However, a stronger and more apparent correlation can be observed when the percentiles across window sizes are aggregated (See Figure 5.10). An interesting observation pertaining to the correlation between the depth of the trees and the window size is illustrated. At shallow tree depth, the larger sized window (15 × 15) based feature positions are evaluated more. While in the middle of the latter tree depth smaller window size based feature positions are evaluated more. This is because, at shallow tree depth, where there are higher uncertainty and more variation in the depth of evaluated superpixels, it is advantageous to evaluate affinity based on larger window sizes. In contrast, at deeper tree levels, smaller window sizes are preferred.

5.3.6 Evaluating Significance of matching costs and window sizes

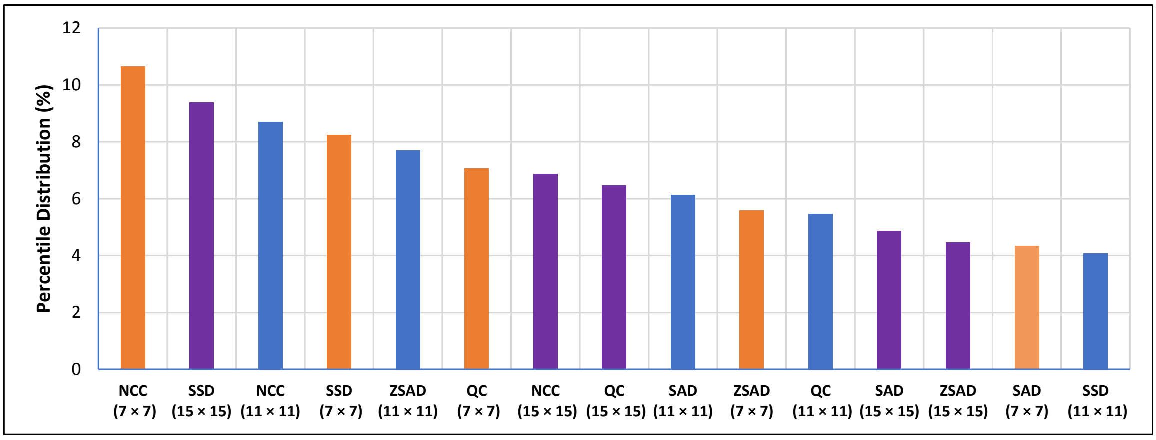


Figure 5.11 Bar chart illustrating an ordered percentile distribution of evaluated features, aggregated based on their window size and matching cost function (SAD, SSD, NCC, ZSAD and QC).

Examining the results presented in Figure 5.10 and 5.13 begs the need for an investigation into the significance of the other features. Since it appears that some features are more evaluated than others and hence are more discriminating. An ordered percentile distribution of evaluated features aggregated based on their window size and matching cost functions from Figure 5.13, is presented in Figure 5.11. It can be hypothesized that the more evaluated a feature is, the more discriminative (and hence more significant) of the target space it is. Consequently, to investigate the effect of the number of features used, the number of features used is progressively decreased based on the level of significance (i.e. the number of times it was evaluated). The results of this are presented in Figure 5.12. The graph illustrates the change in the percentage of correctly predicted pixels (at a threshold of 10mm) as the number of features used is reduced. Note that the order in which the features are dropped is based on the level of their significance according to Figure 5.11, starting with the SSD (11 × 11) feature, followed by the SAD (7 × 7) and so on.

The result indicates that there is a substantial improvement in accuracy as a resulting of using multiple costs functions as hypothesized in the introduction of this chapter. However, it can be argued that the first 8 most significant features would have

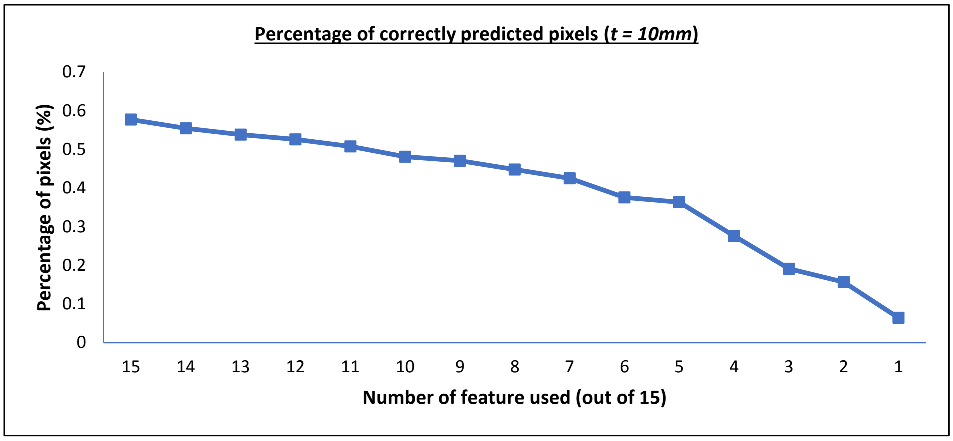


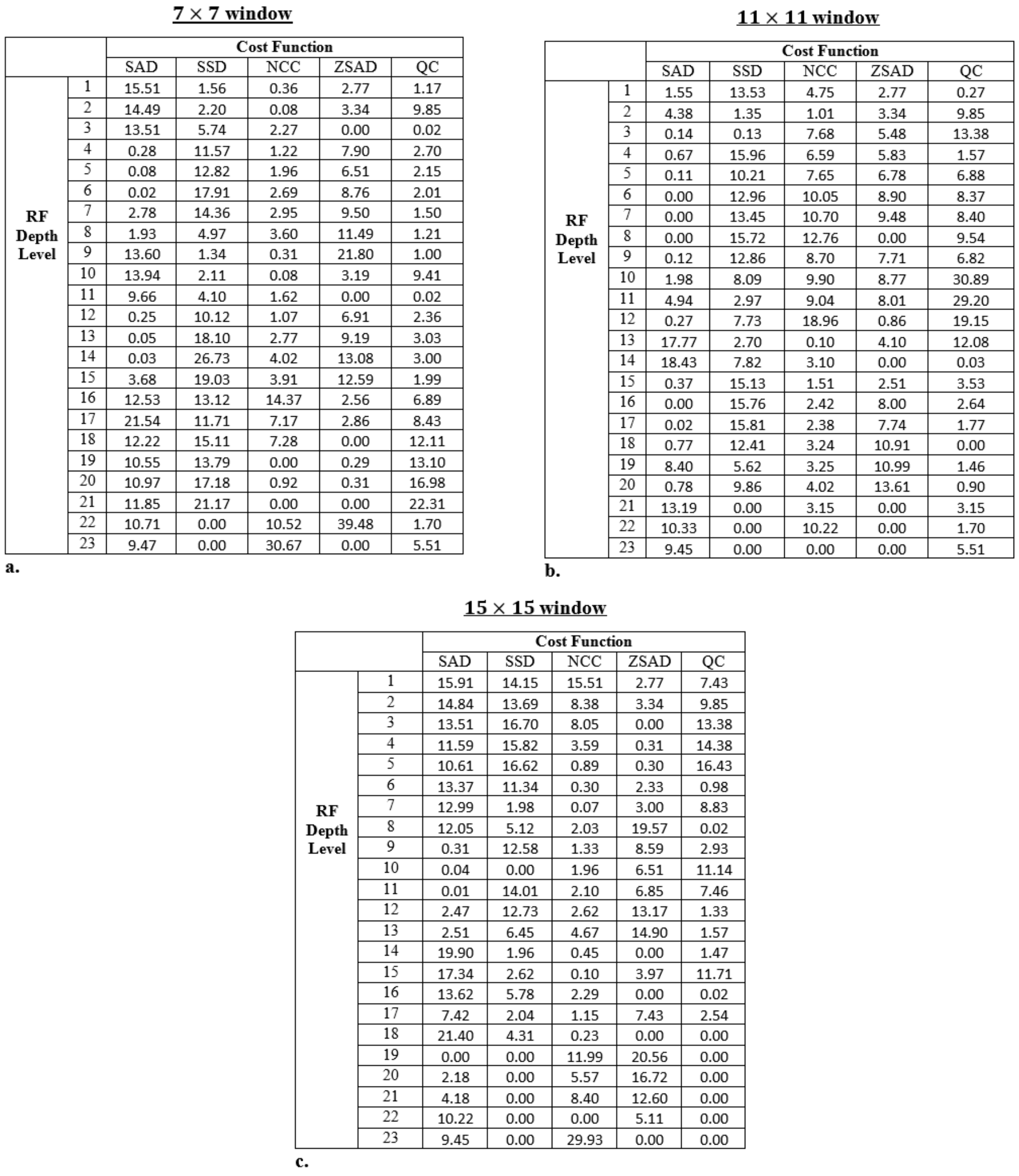
Figure 5.12 Graph illustrating the change in the percentage of correctly predicted pixels (at a threshold of 10mm) as the number of feature used is reduced. Note that the order in which the features are dropped is based on the level of their significance according to Figure 5.11, starting with the SSD (11 × 11) feature before the SAD (7 × 7) and so on.

sufficed as their absence contributes to 87.28%$(\frac{0.577-0.064}{0.448})$ of the drop in the percentage of correctly predicted pixels. This is a potential avenue to improving the efficiency of the proposed framework.

5.4 Summary

In this chapter, an application of the regression forest technique for computing depth from stereo images was proposed and developed. Introducing Conditional Regressive Random Forest, a framework that uniquely combines expert trees based on the features of the superpixel whose depth is being predicted. The framework further enforces smoothness constraints as it predicts the depth of each superpixel away from the camera. Thus, it demonstrates the use of a relatively cheap stereo camera rig to generate a high-quality depth image of the hand that can be used for pose estimation. It should again be noted that the technique is applicable to other scenarios, including regression problems whereby each data point is not purely independent of other data points. In this case, the regressive or classification Random Forest can be applied to independently regress for each data point, whilst, the potential dependency between data points can be modeled by the pairwise term.

An obvious limitation of the proposed technique is the need of a skin segmentation step that precedes the stereo-matching algorithm. Whilst this does not affect the performance of the technique itself, it will affect the shape of the recovered hand depth. False hand segmentation could be an issue in scenarios where the recovered depth is to be used as a feature for further analysis. For instance, in [75] the feature for pose estimation from depth image is dependent on the shape of the hand. Another potential limitation of this technique is that it quantizes the depth space, limiting the depth sensing reach or resolution. Whilst larger depth sensing reach can be learned by adapting the training set appropriately, this will lead to a computation cost vs. depth reach/resolution trade-off. Since larger depth reach or resolution will require more depth levels (and hence increase in the size of the matrices $\boldsymbol{B}$ and $\boldsymbol{Y}$), the



7 × 7 window

		Cost Function				
		SAD	SSD	NCC	ZSAD	QC
RF Depth Level	1	15.51	1.56	0.36	2.77	1.17
	2	14.49	2.20	0.08	3.34	9.85
	3	13.51	5.74	2.27	0.00	0.02
	4	0.28	11.57	1.22	7.90	2.70
	5	0.08	12.82	1.96	6.51	2.15
	6	0.02	17.91	2.69	8.76	2.01
	7	2.78	14.36	2.95	9.50	1.50
	8	1.93	4.97	3.60	11.49	1.21
	9	13.60	1.34	0.31	21.80	1.00
	10	13.94	2.11	0.08	3.19	9.41
	11	9.66	4.10	1.62	0.00	0.02
	12	0.25	10.12	1.07	6.91	2.36
	13	0.05	18.10	2.77	9.19	3.03
	14	0.03	26.73	4.02	13.08	3.00
	15	3.68	19.03	3.91	12.59	1.99
	16	12.53	13.12	14.37	2.56	6.89
	17	21.54	11.71	7.17	2.86	8.43
	18	12.22	15.11	7.28	0.00	12.11
	19	10.55	13.79	0.00	0.29	13.10
	20	10.97	17.18	0.92	0.31	16.98
	21	11.85	21.17	0.00	0.00	22.31
	22	10.71	0.00	10.52	39.48	1.70
	23	9.47	0.00	30.67	0.00	5.51

a.

11 × 11 window

		Cost Function				
		SAD	SSD	NCC	ZSAD	QC
RF Depth Level	1	1.55	13.53	4.75	2.77	0.27
	2	4.38	1.35	1.01	3.34	9.85
	3	0.14	0.13	7.68	5.48	13.38
	4	0.67	15.96	6.59	5.83	1.57
	5	0.11	10.21	7.65	6.78	6.88
	6	0.00	12.96	10.05	8.90	8.37
	7	0.00	13.45	10.70	9.48	8.40
	8	0.00	15.72	12.76	0.00	9.54
	9	0.12	12.86	8.70	7.71	6.82
	10	1.98	8.09	9.90	8.77	30.89
	11	4.94	2.97	9.04	8.01	29.20
	12	0.27	7.73	18.96	0.86	19.15
	13	17.77	2.70	0.10	4.10	12.08
	14	18.43	7.82	3.10	0.00	0.03
	15	0.37	15.13	1.51	2.51	3.53
	16	0.00	15.76	2.42	8.00	2.64
	17	0.02	15.81	2.38	7.74	1.77
	18	0.77	12.41	3.24	10.91	0.00
	19	8.40	5.62	3.25	10.99	1.46
	20	0.78	9.86	4.02	13.61	0.90
	21	13.19	0.00	3.15	0.00	3.15
	22	10.33	0.00	10.22	0.00	1.70
	23	9.45	0.00	0.00	0.00	5.51

b.

15 × 15 window

		Cost Function				
		SAD	SSD	NCC	ZSAD	QC
RF Depth Level	1	15.91	14.15	15.51	2.77	7.43
	2	14.84	13.69	8.38	3.34	9.85
	3	13.51	16.70	8.05	0.00	13.38
	4	11.59	15.82	3.59	0.31	14.38
	5	10.61	16.62	0.89	0.30	16.43
	6	13.37	11.34	0.30	2.33	0.98
	7	12.99	1.98	0.07	3.00	8.83
	8	12.05	5.12	2.03	19.57	0.02
	9	0.31	12.58	1.33	8.59	2.93
	10	0.04	0.00	1.96	6.51	11.14
	11	0.01	14.01	2.10	6.85	7.46
	12	2.47	12.73	2.62	13.17	1.33
	13	2.51	6.45	4.67	14.90	1.57
	14	19.90	1.96	0.45	0.00	1.47
	15	17.34	2.62	0.10	3.97	11.71
	16	13.62	5.78	2.29	0.00	0.02
	17	7.42	2.04	1.15	7.43	2.54
	18	21.40	4.31	0.23	0.00	0.00
	19	0.00	0.00	11.99	20.56	0.00
	20	2.18	0.00	5.57	16.72	0.00
	21	4.18	0.00	8.40	12.60	0.00
	22	10.22	0.00	0.00	5.11	0.00
	23	9.45	0.00	29.93	0.00	0.00

c.

Figure 5.13 Percentile distribution of evaluated features based on their window size and matching cost function (SAD, SSD, NCC, ZSAD and QC) at different depth levels.

computational expense of the technique increases. A solution to this problem might be to use a logarithmic scale for depth so that less resolution will be given to depth prediction far away (which is often more significant) and vice versa. This chapter presented a state-of-the-art machine learning approach in recovering accurate depth images from stereoscopic images of the hand, and both the qualitative and quantitative results show very promising results. This robust depth estimation can be fed into state-of-the-art hand pose (from depth) estimation techniques like those discussed in Chapter 2.

In this chapter, a data-driven Regressive Random Forest framework that learns the direct mapping between a stereo image pair and high-quality groundtruth depth measurement is proposed. In so doing, it presents Conditional Regressive Random Forest (CRRF), an innovative combination of Regressive Random Forest and Conditional Random Fields to model this mapping. A major contribution of this chapter is the use of a machine learning framework to combine various stereo matching criteria (multiple cost functions and window sizes) with the aim of implicitly determining stereo correspondences. Unlike conventional CRF methods that require iterative solutions, a closed form solution to CRRF inference was derived.

A comment on the achieved results is that these are based on the assumption that a perfectly correct ground truth was acquired. Of course, this is not the case as the ground truth is based on re-projected depth measurement from the RGBD sensor. The re-projection (based on the pre-calibration), as well as, the RGBD sensor itself contributes to an erroneous ground truth. A more reliable approach to the data collection (for training, as well as, validation) could involve laser-based depth sensors that are more accurate and less susceptible to background noise like the RGBD sensor used in this thesis. It should be noted that although a laser is an effective way to get high accuracy, it is really expensive.

So far in the thesis, the premise has been to first recover depth (from a hand scene) with the aim of later using the resolved depth to estimate hand pose. The work presented so far will suffice in testing this premise as there is substantial work in literature for robust pose recovery from depth. Hence the task is limited to applying these frameworks to the recovered hand depth. This chapter concludes the investigation into this premise. The next chapter will explore the possibility of implementing a unified framework that recovers pose directly from stereo. A comparison of these two approaches (i.e. recovering depth explicitly for later pose estimation or directly recovering pose from stereo) will be investigated in the latter part of the thesis.

Chapter 6

Hand Pose Estimation Using Deep Stereovision and Markov-chain Monte Carlo

As discussed at the end of the previous chapter, the approach taken so far in this thesis is to recover robust hand depth with the aim of later using the recovered depth to predict the pose. This chapter presents an alternative approach in that it proposes a solution to estimating pose directly from stereo. The proposed framework combines jointly optimal depth and hand pose estimation in a unified framework using Markov-chain Monte Carlo sampling and deep learning. Inspired by the work of Collins and Carr in [113], this chapter presents a joint optimization solution to depth recovery and pose estimation from stereo capture. In [113] a hybrid stochastic/deterministic optimization scheme that uses Reverse Jump Markov-chain Monte Carlo is used to perform stochastic search over a space of potential object detections. This stochastic search is interleaved with deterministic-based optimization of association of these proposed detections in along succeeding frames. In this chapter MCMC is applied to propose depth images that are tested against observed stereo information and prior probability to estimate the hand pose.

In hand pose estimation, the aim is to regress for the spatial location of the different hand joints given a pair of images from a stereo capture of the hand. With this in mind, it is important to recognize the success of depth based (compared to RGB based) hand pose estimation, hence it is useful to exploit depth as a hidden variable between a stereo image input and the spatial pose output. To this end, the problem is conceptualized to jointly solving for two unknowns: the depth image and the spatial pose of hand joints. Unlike several approaches to pose estimation from stereo capture, which explicitly recovers disparity before regressing for the pose in a sequential manner, this chapter presents a joint optimization approach that is robust against potential errors in the depth estimation. Thus, this reduces the burden on the pose estimation framework to be robust against erroneous depth recovery. The consequence of this approach is that it iteratively revises for errors in depth proposal. This allows for simultaneous correction of proposed depth estimation and the resulting pose estimation to jointly optimize the likelihood of the depth and hand pose estimation given the stereo input. **Contributions**: Unlike several approaches to pose estimation from stereo capture that explicitly recover disparity before regressing for the pose in a sequential manner, this chapter presents a joint optimization approach that is robust against potential error in the depth estimation pre-step. Thus, there is no burden on the pose estimation framework to be robust against erroneous depth recovery.

Another consequence of the proposed approach is that it iteratively revise for errors in depth proposal. This allows for simultaneous correction of proposed depth estimation and the resulting pose estimation to jointly optimize the likelihood of the depth and hand pose estimation given the stereo input.

Lastly, unlike the work in [64], which utilizes a state-of-the-art tracking method that is sensitive to erroneous initialization and anatomical hand size as discussed in [82], the approach proposed is a semi-generative approach that is experimentally proven to work on different sizes and tones of hand without pre-calibration.

6.1 Methodology

As discussed in the previous section, depth is a strong hidden variable in the context of the input domain, stereo capture; and output domain, pose estimation. This is a consensus that most work on pose estimation from stereo has made ([64, 85, 86]), a concept which is the foundation of this proposed work.

6.1.1 Stereo-Depth-Pose

For a given stereo image pair, $\boldsymbol{S}$ of a scene of a hand pose $\boldsymbol{H}$, with a depth image $\boldsymbol{D}$, it is assumed that the hand pose induces a depth surface, that in turn induces the detected stereo image in a Bayesian tree model. See Figure 6.1a. The goal is then reduced to establishing the pose $\boldsymbol{H}^*$ and depth, $\boldsymbol{D}^*$ that maximize the posterior distribution of $\boldsymbol{H}$ and $\boldsymbol{D}$ given an observed stereo image pair $\boldsymbol{S}$.

$$\boldsymbol{H}^*, \boldsymbol{D}^* = \underset{\boldsymbol{H},\boldsymbol{D}}{\arg\max}\, Pr(\boldsymbol{H}, \boldsymbol{D}|\boldsymbol{S}) \tag{6.1}$$

Following from the Bayesian tree model above, it can be assumed that $\boldsymbol{H}$ and $\boldsymbol{S}$ are conditionally independent, given $\boldsymbol{D}$. This implies

$$Pr(\boldsymbol{S}, \boldsymbol{H}|\boldsymbol{D}) = Pr(\boldsymbol{S}|\boldsymbol{D})Pr(\boldsymbol{H}|\boldsymbol{D}) \tag{6.2}$$

and

$$Pr(\boldsymbol{S}|\boldsymbol{H}, \boldsymbol{D}) = Pr(\boldsymbol{S}|\boldsymbol{D}). \tag{6.3}$$

From Bayes' theorem, one can infer that

$$Pr(\boldsymbol{S}|\boldsymbol{H}, \boldsymbol{D}) = \frac{Pr(\boldsymbol{H}, \boldsymbol{D}|\boldsymbol{S})Pr(\boldsymbol{S})}{Pr(\boldsymbol{H}, \boldsymbol{D})}, \tag{6.4}$$

and that given Eq. 6.3 and Eq. 6.4 it can be inferred that

$$Pr(\boldsymbol{H}, \boldsymbol{D}|\boldsymbol{S}) = \frac{Pr(\boldsymbol{S}|\boldsymbol{D})Pr(\boldsymbol{H}, \boldsymbol{D})}{Pr(\boldsymbol{S})}. \tag{6.5}$$

Note $Pr(\boldsymbol{H}, \boldsymbol{D}) = Pr(\boldsymbol{H}|\boldsymbol{D})P(\boldsymbol{D})$, therefore from Eq. 6.5 one can see that

$$Pr(\boldsymbol{H}, \boldsymbol{D}|\boldsymbol{S}) = \frac{Pr(\boldsymbol{S}|\boldsymbol{D})Pr(\boldsymbol{H}|\boldsymbol{D})Pr(\boldsymbol{D})}{Pr(\boldsymbol{S})}. \tag{6.6}$$

and hence Eq. 6.1 can be represented as

$$\boldsymbol{H}^*, \boldsymbol{D}^* = \arg\max_{\boldsymbol{H},\boldsymbol{D}} Pr(\boldsymbol{S}|\boldsymbol{D})Pr(\boldsymbol{H}|\boldsymbol{D})Pr(\boldsymbol{D}). \tag{6.7}$$

The posterior joint probability of $\boldsymbol{H}$ and $\boldsymbol{D}$ yields a very high dimensional space. An intuitive solution to this joint probability will be to first determine the depth image, $\boldsymbol{D}^*$ that best describes the observed stereo image pair $\boldsymbol{s}$,

$$\boldsymbol{D}^* = \arg\max_{\boldsymbol{D}} Pr(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{D}) \tag{6.8}$$

before using $\boldsymbol{D}^*$ to determine the corresponding pose,

$$\boldsymbol{H}^* = \arg\max_{\boldsymbol{H}} Pr(\boldsymbol{H}|\boldsymbol{D}^*)Pr(\boldsymbol{D}^*). \tag{6.9}$$

This is the approach of several papers on hand pose estimation from stereo capture, including [64]. Recall that the aim was to first establish a robust depth image given a stereo image capture that can then be used to predict the hand pose. However, this does not fully optimize the pose-depth joint probability space. This is because it assumes that the depth that maximizes $Pr(\boldsymbol{D})$ coincides with the point (i.e. the pose and depth image) that maximizes in the pose-depth joint distribution. This is not always the case. Consider Figure 6.1b, where a hypothetical joint distribution between $\boldsymbol{H}$ and $\boldsymbol{D}$ is presented for a given stereo image pair. The maximum probability is indicated with the red dot. Consider a case where the joint distribution is first

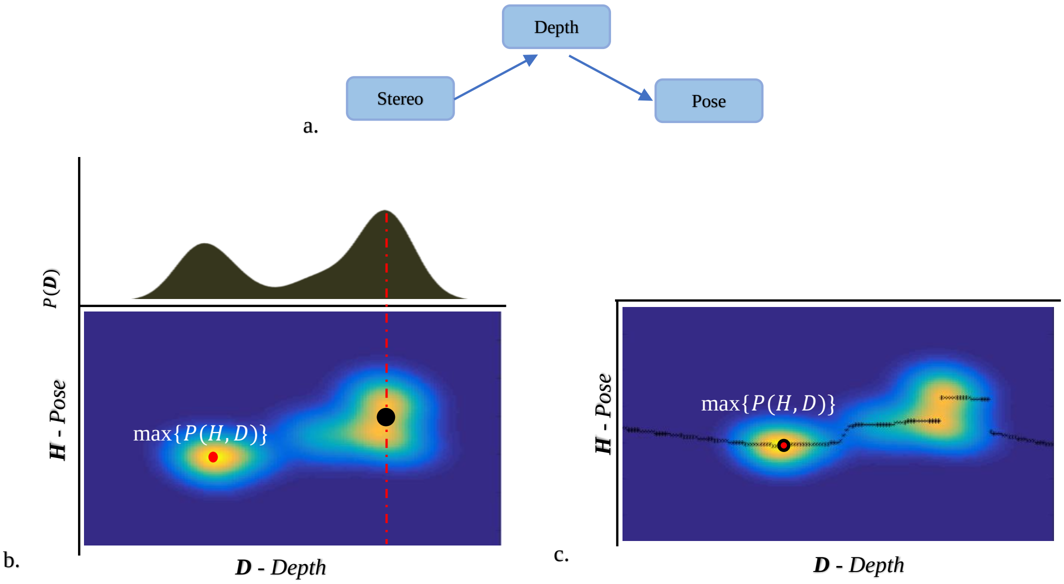


Figure 6.1 (a) Bayesian tree model of the relationship between depth, stereo images, and hand pose in the proposed model. (b) Illustrates a conventional approach to estimating pose from stereo capture, where the optimum depth is first determined and then used to factorize the joint probability to identify the maximizing pose. (c) The proposed stochastically evaluates potential depth solutions (along the black line) and then a maximizing pose is established. This will guarantee identifying the joint maximum point (illustrated with the red dot) with enough depth proposals.

marginalized along $\boldsymbol{H}$ for the depth probability, $Pr(\boldsymbol{D}) = \sum_{\boldsymbol{H}} Pr(\boldsymbol{H}, \boldsymbol{D})$ to identify $\boldsymbol{D}^*$(analogous to solving for a robust depth from a given a stereo image pair as in Eq. 6.8). $\boldsymbol{H}^*$ is then determined by maximizing $Pr(\boldsymbol{H}|\boldsymbol{D}^*)$, illustrated with the red dotted line (analogous to Eq. 6.9). Note how the optimized maximum does not coincide with the joint maximum. Secondly, it assumes that the depth image computed from the stereo image is fully correct or else even more robust and complex pose estimation from depth techniques will be required to handle erroneous depth recovery. Inspired by [113], a different approach is taken. Instead, in this research a search for the optimum $\boldsymbol{D}^*$ along the manifold described by the optimum $\boldsymbol{H}$ for all potential depth images is done, as in:

$$\boldsymbol{D}^* = \arg\max_{\boldsymbol{D}} [\phi_H Pr(\boldsymbol{S}|\boldsymbol{D})], \tag{6.10}$$

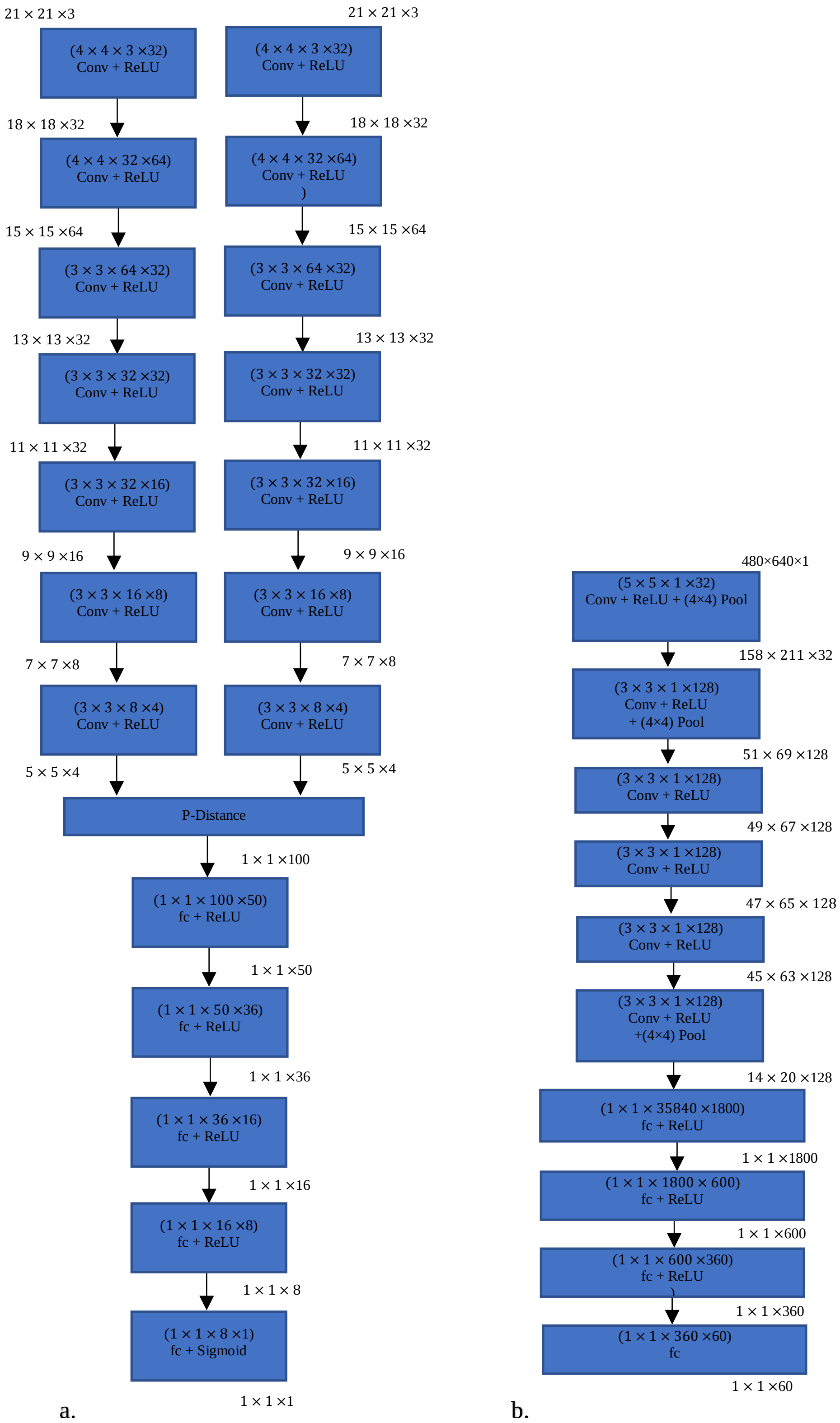


Figure 6.2 Structure of the two CNNs used. (a). A Siamese network is used as a similarity measure between two potentially matching square patches of pixels. (b) illustrates the structure used for discriminatively regressing for pose given a depth image.

where

$$\phi_H = \max_{\boldsymbol{H}}\{Pr(\boldsymbol{H}|\boldsymbol{D})Pr(\boldsymbol{D})\}, \tag{6.11}$$

and in turn, $\boldsymbol{H}^*$ is computed using Eq. 6.9. Note the effect of this as shown in Figure 6.1c, where the manifold is illustrated with a black line. Consequently, the high dimensional space of the depth and the pose is iteratively explored by proposing a depth and evaluating for Eq. 6.6 in search of a maximum.

6.1.2 Probability of Observed Stereo Image given Proposed Depth

To efficiently propose a depth image, first the reference stereo image is segmented as in the previous chapter into superpixels using Simple Linear Iterative Clustering (SLIC) [107]. A hand depth image is represented with a vector, $\boldsymbol{d}$, of the depth values of all the superpixels that lie within the hand region. Henceforth, this vector will be referred to as the *depth configuration vector*. For a proposed depth image,

$$Pr(\boldsymbol{S}|\boldsymbol{D}) = \log\left\{\prod_{j}^{J} Pr(\boldsymbol{S}|d_j)\right\} = \sum_{j}^{J}\log[Pr(\boldsymbol{S}|d_j)] \tag{6.12}$$

where there are J hand superpixels. The probability of a stereo image pair given the depth of the j^{th} superpixel, $Pr(\boldsymbol{S}|d_j)$, is modeled as the re-projection affinity of the proposed d_j. For a proposed depth, the intrinsic and extrinsic parameters of the stereo rig are used to reproject pixels in the reference stereo image plane onto the corresponding image plane, before computing affinity. See Section 3.1.3 for details. The quality of a proposed depth is evaluated based on how reprojected superpixels match the original superpixel. Hence, for a stereo image pair with superpixel x_j in the left image with a centroid pixel position $\begin{bmatrix} x_L^j \\ y_L^j \end{bmatrix}$ and a proposed depth d_j,

$$Pr(\boldsymbol{S}|d_j) = C\Big(I_L(x_L^j, y_L^j), I_R(x_R^{*j}, y_R^{*j})\Big) \tag{6.13}$$

where $C(,)$ is a window-based matching cost function that gives a measure of affinity and

$$\begin{bmatrix} x_R^* \\ y_R^* \end{bmatrix} = F\left(\begin{bmatrix} x_L \\ y_L \end{bmatrix}, d\right) = d \begin{bmatrix} x_L \\ y_L \\ 1 \end{bmatrix} \boldsymbol{P}_L^{-1}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{P}_R. \tag{6.14}$$

Here $\boldsymbol{P}_L$ and $\boldsymbol{P}_R$ are the projection matrices of the left and right stereo camera pair and $\boldsymbol{R}$ and $\boldsymbol{t}$ are the relative extrinsic matrix and vector respectively established for the stereo camera using camera calibration (see Chapter 2 and [103] for more details). $C(,)$ is represented as a Siamese network, as in the work of Zbontar and LeCun, [114]. Deep Siamese networks have recently become a popular method to establish similarity in state-of-the-art stereo matching algorithms. The Siamese network architecture presented in [114] was used as it has been proven to have very strong stereo-matching performance (based on a very high Middlebury dataset ranking). The first subnet consists of a pair of layers, each composed of a convolutional layer followed by a ReLU, as shown in Figure 6.2a. This is followed by the P-Distance layer that computes the square distance of each feature vector in one of the pair of subnets to the other. Finally followed by four fully connected (fc) and ReLU layers; and then a fully connected and sigmoid layer.

The output of the sigmoid layer, which ranges from 0 to 1, is the similarity score $C(,)$. Hence the probability of the observed stereo image, $\boldsymbol{S}$, given a proposed depth configuration, $Pr(\boldsymbol{S}|\boldsymbol{D})$, is modeled as the similarity of the disparity correspondence resolved from the proposed depth.

6.1.3 Probability of Pose Conditioned on Depth

The second component is the probability of the pose $\boldsymbol{H}$ given depth $\boldsymbol{D}$. Note that the ultimate task is to establish ϕ_H. For ease of implementation ϕ_H is redefined as

$$\phi_H = Pr(\boldsymbol{H} = \boldsymbol{h})Pr(\boldsymbol{D}), \tag{6.15}$$

with

$$\boldsymbol{h} = \arg\max Pr(\boldsymbol{H}|\boldsymbol{D}). \tag{6.16}$$

$Pr(\boldsymbol{H} = \boldsymbol{h})$ is the probability of a unique pose, $\boldsymbol{h}$, based on the hand pose prior distribution (described in Section 6.1.4). Hence a discriminative model that determines the optimum pose $\widehat{\boldsymbol{H}}$ of a given depth, $\boldsymbol{D}$ is applied here. Note that $\widehat{\boldsymbol{H}}$ is only the solution to a single proposed depth at proposal iteration. The probability value, ϕ_H, is then estimated as the probability of this pose solution, $\widehat{\boldsymbol{H}}$. The assumption made here is that the discriminatively determined pose, $\widehat{\boldsymbol{H}}$, is the pose that maximizes the posterior, $\arg\max_{\boldsymbol{H}} Pr(\boldsymbol{H}|\boldsymbol{D})$, and that $Pr(\boldsymbol{H} = \widehat{\boldsymbol{H}})$ is the maximum posterior probability, $\max_{\boldsymbol{H}} Pr(\boldsymbol{H}|\boldsymbol{D})$. The idea here is that after several iteration of depth proposals the, $\widehat{\boldsymbol{H}}$ should tend to the optimum pose solution, $\boldsymbol{H}^*$ for the observed stereo capture, $\boldsymbol{S}$

The discriminative model used here is also a CNN. Henceforth, this second CNN will be referred to as the *pose-estimation network*. The pose-estimation network takes a [640 × 480] single channel depth image (from the proposed depth configuration) and outputs a $3 * K$-dimensional vector that represents the 3D spatial coordinates of all K joints that describe a hand pose. So, in effect, for a given depth image, the pose-estimation network computes a single pose. ϕ_H is the product of the probability of the estimated pose (based on the pose prior, $Pr(\boldsymbol{H})$) and the probability of the given depth image (based on the depth image prior, $Pr(\boldsymbol{D})$). Both priors are described in the following subsection. The structure of the pose-estimation network is illustrated in Figure 6.2b. This consists of six convolutional layers (each followed with a ReLU). Three of these also with a Pooling layer followed by four fully connected layers (each followed by a ReLU layer except the last). The output of the final fully connected layer indicates the joint positions. This network structure is inspired by the work of Oberweger et al. in [5], where different CNN architectures had been tested for hand pose estimation from a depth image. The architecture used in this chapter was chosen because of its performance and simplicity.

6.1.4 Prior over Depth and Pose

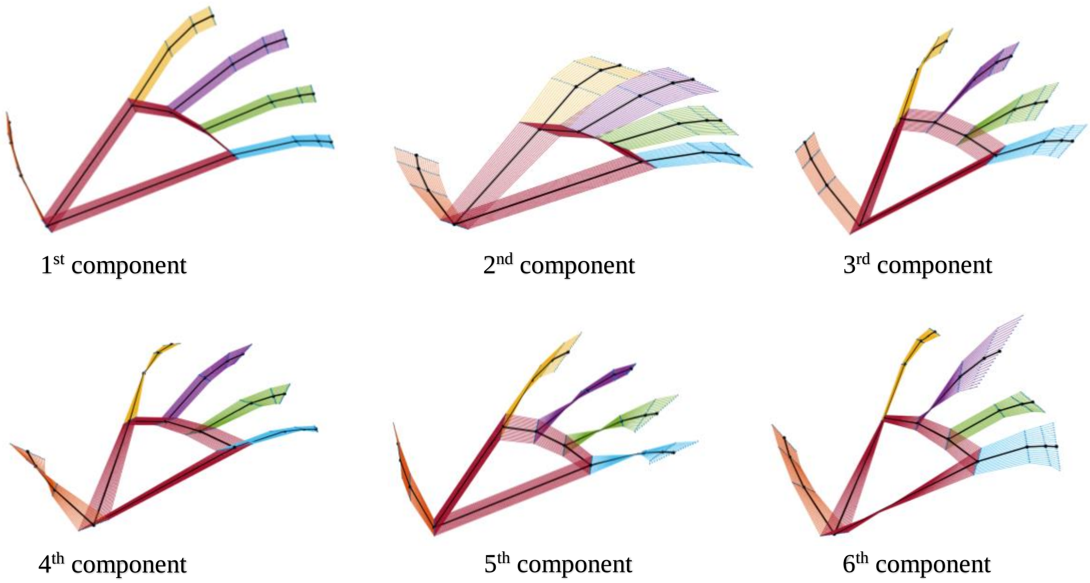
Pose: Let $\boldsymbol{h}$ denote the hand pose vector in a $3*K$-dimensional space $\boldsymbol{V}$. To establish a pose prior over the hand, a constraint that determines the joint configuration, a member of a subspace, $\boldsymbol{W} \subset \boldsymbol{V}$ is added. Where $\boldsymbol{W}$ is a subspace that potentially contains all the possible hand poses or at least all the poses observed in a comprehensive dataset of hand pose. During implementation, Dataset C presented in Section 3.3.3 was used. To mathematically represent this dataset, a criterion for $\boldsymbol{W}$ was established, based on the principal components that span all poses in the dataset of prior poses. Applying principal component analysis (PCA) on the prior pose dataset, the N most significant components were established, $\boldsymbol{E} = [\boldsymbol{e}_1, ..., \boldsymbol{e}_N]$, where $N << 3 * K$. More formally, consider Dataset C with J data captures, represented by $\{(Z_S^1, i_D^1, \boldsymbol{h}_H^1), ..., (Z_S^J, i_D^J, \boldsymbol{h}_H^J)\}$, where $Z_S^j = \{z_j, z'_j\}$ is the j^{th} stereo capture pair; i_D^j is the j^{th} depth image and $\boldsymbol{h}_H^j$is the j^{th} pose vector. A database of prior poses is established as in $\{\boldsymbol{h}_H^1, \boldsymbol{h}_H^2, .., \boldsymbol{h}_H^J\}$. The elements $\boldsymbol{e}_1, ..., \boldsymbol{e}_{3*K}$ are the ordered eigenvectors of $Cov([\boldsymbol{h}_H^1, \boldsymbol{h}_H^2, .., \boldsymbol{h}_H^J])$, where $Cov()$ returns the covariance matrix of its argument. To maintain realistic hand pose prediction a constraint that a newly computed pose $\boldsymbol{h}$ should be represented by a linear combination of the established component, $\boldsymbol{h}' - \boldsymbol{\mu} \approx \sum_i^N a_i \boldsymbol{e}_i$, is applied. Where $\boldsymbol{\mu}$ denotes the mean pose of all joint configurations in the prior dataset and a_i is the weighting assigned to the i^{th} eigenvector, $\boldsymbol{e}_1$. To this end, the probability of a resolved pose $\boldsymbol{h}'$ is established as

$$Pr(\boldsymbol{H} = \boldsymbol{h}') = e^{||\boldsymbol{E}\boldsymbol{\alpha}^* + \boldsymbol{\mu} - \boldsymbol{h}'||} \tag{6.17}$$

where

$$\boldsymbol{\alpha}^* = \boldsymbol{E}^+(\boldsymbol{h}' - \boldsymbol{\mu}) \tag{6.18}$$

and $||.||$ denotes the l^2-norm and $\boldsymbol{E}^+$ is the pseudo-inverse of the $\boldsymbol{E}$. $\boldsymbol{\alpha}^*$ is then the least square estimation to the coefficients of the components that yield $\boldsymbol{h}'$ under a linear combination. Then the exponentiated Euclidean distance between this linear



1st component

2nd component

3rd component

4th component

5th component

6th component

Figure 6.3 The variations captured by the first six components of $\boldsymbol{E}$. The mean pose is represented in black.

combination of components and $\boldsymbol{h}'$ is used as a measure of the prior probability. In effect, a 3D joint configuration (pose) that is like those in the dataset will be more accurately mapped onto $\boldsymbol{W}$ and remapped back. To illustrate the information captured by these components, the variation from the mean based on the first six eigenvectors (see Figure 6.3). Note how the first and second component capture rigid spatial displacement in the dataset whilst the succeeding component captures more nuanced variations in the poses.

Depth: Using the hand region segmentation, the Euclidean distance between the mean hand pixel position in both images of the stereo pair is used to estimate the general distance of the hand to the camera, using the baseline and focal lengths of the stereo rig. The prior over depth at all superpixels in the scene is modeled with a Gaussian, with a mean as the estimated general distance, R and a standard deviation, σ, as in

$$Pr(\boldsymbol{D} = \boldsymbol{d}) = \sum_j \frac{a}{\sigma\sqrt{2\pi}} e^{\frac{-(d_j - R)^2}{2\sigma^2}} \tag{6.19}$$

The discussion so far introduces the idea of a proposing depth configuration that is consistent with the observed stereo capture and that yield poses that are realistic (i.e. similar to those in the dataset). The part of the discussion that is yet to be discussed is the framework for proposing this depth configuration. A Markov-chain Monte Carlo framework is used. As introduced in Section 3.2.7, MCMC is particularly useful in scenarios where it is required to sample from a high dimensional space with a potentially complex probability distribution. This is clearly the case here, in that the number of hand regions (i.e. the length of the depth configuration vector) is relatively large (ranging from 30 to 70 in the experiment below) and the distribution along these dimensions is not a simple one. The MCMC framework used is presented in the next section.

6.1.5 Metropolis-Hastings Algorithm

To achieve an informed framework for proposing depth images (configuration), Markov-chain Monte Carlo is exploited. More specifically the Metropolis-Hastings Algorithm (in Algorithm 1) is used to determine a set of depth proposals in an iterative manner. To this end, a new depth configuration $\boldsymbol{D'}$ is proposed based on a distribution that is conditioned on the previous proposal $q(\boldsymbol{D'}, \boldsymbol{D}^{(i)})$. This distribution is implemented by randomly perturbing the elements of the depth vector $\boldsymbol{d}^{(i)}$ that describes $\boldsymbol{D}^{(i)}$. In this implementation, the perturbation was selected randomly within a range of 0 to $10mm$. This was empirically chosen during implementation. Hence the probability of the newly proposed depth vector, $\boldsymbol{d'}$ is dependent on the previous depth proposal $\boldsymbol{d}^{(i)}$. Subsequently, given the newly proposed depth, and an acceptance ratio, α, where

$$\alpha(\boldsymbol{D'}, \boldsymbol{D}^{(i)}) = \min\left\{1, \frac{Pr(\boldsymbol{D'}, \boldsymbol{H'}|\boldsymbol{S})}{Pr(\boldsymbol{D}^{(i)}, \boldsymbol{H}^{(i)}|\boldsymbol{S})}\right\}, \tag{6.20}$$

and

$$\boldsymbol{H'} = \max_{\boldsymbol{H}} Pr(\boldsymbol{H}, \boldsymbol{D'}). \tag{6.21}$$

Algorithm 1: Joint Depth and Pose Estimation using the Metropolis-Hastings Algorithm

```
Input: $\boldsymbol{S}$;
Output: $\boldsymbol{H}^*$;
Initialize $\boldsymbol{D}^{(0)}, \boldsymbol{H}^{(0)} = \arg\max_{\boldsymbol{H}} Pr(\boldsymbol{H}|\boldsymbol{D}^{(0)})$;
Let $\boldsymbol{D}^* = \boldsymbol{D}^{(0)}, \boldsymbol{H}^* = \boldsymbol{H}^{(0)}$;
for $i = 0$ to $L - 1$ do
    Sample $u \sim U_{[0,1]}$ ;
    Sample $\boldsymbol{D}' \sim q(\boldsymbol{D}'|\boldsymbol{D}^{(i)})$ ;
    if $\log(u) < \log(\alpha(\boldsymbol{D}'|\boldsymbol{D}^{(i)}))$ then
        $\boldsymbol{D}^{(i+1)} = \boldsymbol{D}'$;
        $\boldsymbol{H}^{(i+1)} = \arg\max_H Pr(\boldsymbol{H}|\boldsymbol{D}^{(i+1)})$;
    else
        $\boldsymbol{D}^{(i+1)} = \boldsymbol{D}^{(i)}$;
        $\boldsymbol{H}^{(i+1)} = \boldsymbol{H}^{(i)}$;
    end
    if $Pr(\boldsymbol{H}^{(i+1)}, \boldsymbol{D}^{(i+1)}|\boldsymbol{S}) > Pr(\boldsymbol{H}^*, \boldsymbol{D}^*|\boldsymbol{S})$ then
        $\boldsymbol{D}^* = \boldsymbol{D}^{(i+1)}$;
        $\boldsymbol{H}^* = \boldsymbol{H}^{(i+1)}$;
    end
end
```

Note that the ratio of the probability of proposing a particular $\boldsymbol{D}'$ given $\boldsymbol{D}^{(i)}$, $q(\boldsymbol{D}', \boldsymbol{D}^{(i)})$, and the reverse is ignored, as these are equal and hence cancel out. If the acceptance ratio is higher than u, a sample between 0 and 1, the proposed depth configuration and the corresponding maximizing pose are considered as a potential candidate for the solution. Hence the higher the probability of the newly proposed depth configuration (relative to the previous proposal) the higher the likelihood that it would be accepted as a potential solution. All potential solutions are evaluated by maximizing for Eq. 6.5. See above for the pseudo-code of the Metropolis-Hastings Algorithm. The effect of this is that the depth configuration that is most consistent with the observed stereo capture and that yields the more probable pose is evaluated for, from within a sample set with a distribution that is consistent with the solution.

6.1.6 Unified Framework

So far all the different components of the framework have been introduced. This section recaps all the components presented and reiterates the description of the entire framework and how a potential depth solution is proposed. The depth is evaluated in two paths. First, it is used to discriminatively compute a pose, the likelihood of this pose is evaluated based on prior pose knowledge. Secondly, the depth is evaluated against the observed stereo capture (using the similarity network). These two probabilities are combined as a representation of the likelihood of the proposed depth. The Metropolis-Hastings algorithm is used to make these proposals in the MCMC sampler. See Figure 6.4.

For greater clarity, the entire framework is summarised, identifying and outlining key features and how they relate in Table 6.1. The next section gives some more details on the implementation of the system described above.

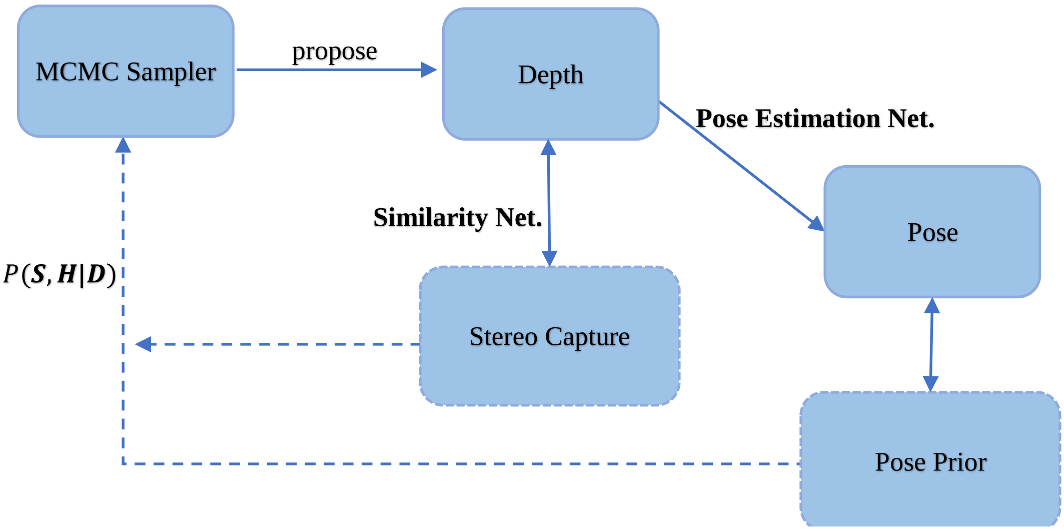


Figure 6.4 An illustration of the MCMC proposal approach. A potential depth solution is proposed by the sampler. This depth is evaluated in two paths. First, it is used to discriminatively compute (using the pose estimation network) for a pose, the likelihood of this resolved pose is evaluated based on prior pose knowledge. Secondly, the depth is evaluated against the observed stereo capture (using the similarity network). These two probabilities are combined to yield the validity of the proposed depth and also used to inform the next depth proposal.

Components	**Implication**
MCMC Sampler	The MCMC sampler proposes different depth configurations (depth images) with a frequency distribution that is consistent with the likelihood of the depth. Hence more probable depth solutions are proposed with more frequency than less probable depth solutions.
Similarity Network	A Siamese network that evaluates the similarity of two image patches. Hence for a given depth solution, points in the reference stereo image pair can be reprojected onto the other stereo image pair plane. The affinity of the reprojected location (in the second stereo image) to the original location in the reference stereo image can be evaluated with the Similarity Network.
Pose Estimation Network	A CNN that determines the hand pose given a proposed depth solution.
Pose Prior	A probability prior over pose whereby poses that are similar to those in the dataset are assigned higher probability.
Depth Prior	This is the prior distribution of the depth values based on the average shift of hand region pixels in the stereo camera pair.

Table 6.1 A brief outline of key components of the proposed framework. This includes MCMC Sampler, Similarity Network, Pose Estimation Network, Pose Prior and Depth Prior.

6.2 Implementation Details

Both the pose-estimation and similarity networks were implemented using the VLFeat MatConvNet [115] and trained on the NVIDIA Titan X GPU with 12GB memory.

Similarity Network: This CNN was trained with the learning rate of 0.001. 10 epochs were executed, reducing the learning rate by 10% with every epoch. The decay, weight, and momentum were set as 0.0005 and 0.09 respectively. Like [114], the similarity network was trained to map a pair of window regions $< I_L(\boldsymbol{p}), I_R(\boldsymbol{q}) >$ from the left and right stereo pair to a cost, c. Hence, the training dataset consists of a pair of potentially matching patches and a target value t as in $\{(< I_L(\boldsymbol{p}_1), I_R(\boldsymbol{q}_1) >, t_1), ..., (< I_L(\boldsymbol{p}_K), I_R(\boldsymbol{q}_K) >, t_J)\}$. This dataset consists of positive (i.e. a matching input window pair) and negative (i.e. a non-matching input window pair) data sample pairs such that

$$\boldsymbol{q}_j = \boldsymbol{p}_j - \boldsymbol{\delta}_j + \boldsymbol{\epsilon}_k \tag{6.22}$$

where $\boldsymbol{\delta}_j$ is the groundtruth disparity shift at the centroid pixel of superpixel j and $\boldsymbol{\epsilon}_k \in$ [minimum Disparity, maximum Disparity] is the randomly assigned shift value. Consequently, a data sample k where $\boldsymbol{\epsilon}_k = 0$ is considered a positive data sample and vice versa otherwise. Hence the target t_k is set to 1 in the case of a positive sample and 0 otherwise. Training is based on a hinge loss function, $max(0, g + c_- + c_+)$, where g is the margin; c_- is the output of the CNN from a non-matching input window patch pair; and c_+ is the output of the CNN from a matching input window patch pair. For each superpixel, a square window region centered on its centroid pixel is considered as the first patch in the reference stereo image. The groundtruth corresponding patch is established by reprojection based on the camera parameters of the stereo cameras and the groundtruth depth at the superpixel. The value of g was set to 0.2 and the maximum and minimum disparity were chosen to be 5 and 48 respectively. This pixel shift range was chosen based on the dataset used i.e. the minimum and maximum hand pixel shift.

Pose-Estimation Network: The pose-estimation network has a significantly greater number of weights due to the larger input image size. To reduce the number of connections (weights) between layers, the max pooling (in the pooling layer) is applied. This was not required in the similarity network due to its relatively small input image. The pose-estimation network was trained with a learning rate of 0.00001 for 150 epochs. Decay weight and momentum were set as 0.005 and 0.09 respectively. Training was done under a mean squared error between the output vector and the groundtruth pose vector. The pose vector consisted of 60 elements i.e. the hand pose was represented with the spatial location of 20 hand joints, $K = 20$. These included the fingertip, distal phalanges, intermediate phalanges, and proximal phalanges of each of the five fingers (see Section 3.5 for more detail).

The prediction phase of the entire framework for a frame of stereo images under 200 MCMC proposals took 360 seconds.

6.3 Experimental Results

The approach was validated experimentally, producing both qualitative and quantitative results. Four main comparisons were made, these included: pose estimation predictions made from single shot depth recovery; estimation made without the pose prior; estimation made using ELNF (Chapter 4); and estimation made using depth acquired using active RGBD camera sensor by computing the percentage of correctly predicted joint positions, $\frac{\sum_{p \in N} \mathbb{I}[|\boldsymbol{s}_p^{GT} - \boldsymbol{s}_p| < G]}{N}$, where $\boldsymbol{s}_p^{GT}$ and $\boldsymbol{s}_p$ are the groundtruth and the predicted 3D joint position of all joints, p in the testing dataset; G is the varying threshold that determines the corrected predicted pixels and N is the total number of joints evaluated (across all the frames). The mean distance error, $\frac{1}{N} \sum_{p \in N} |\boldsymbol{s}_p^{GT} - \boldsymbol{s}_p|$ is also computed so as to quantitatively evaluate the performance of the test. Comparison with the work in work presented in Chapter 6 is presented in the next chapter.

6.3.1 Dataset

As previously described, the Dataset C in Section 3.3.3 was used for training and validating the work. To train the similarity network, a binary class dataset was created with matching pairs of image patches (from the left and right stereo image) as a positive class or considered non-matching otherwise. In the case of the pose-estimation network, data from 2 participants (out of 12) was reserved for testing, and the remaining data (from the other ten participants) was used for training in a cross-validation manner. SLIC segmentation was applied to all reference stereo images, producing approximately 300 superpixels per image. Note that only a fraction of these 300 superpixels were hand region superpixels. The number of hand superpixels (ranging approximately from 30 to 70 per image capture) depends on the distance from the camera and the size of the hand. All in all, about 540,000 patches were used in training the similarity network.

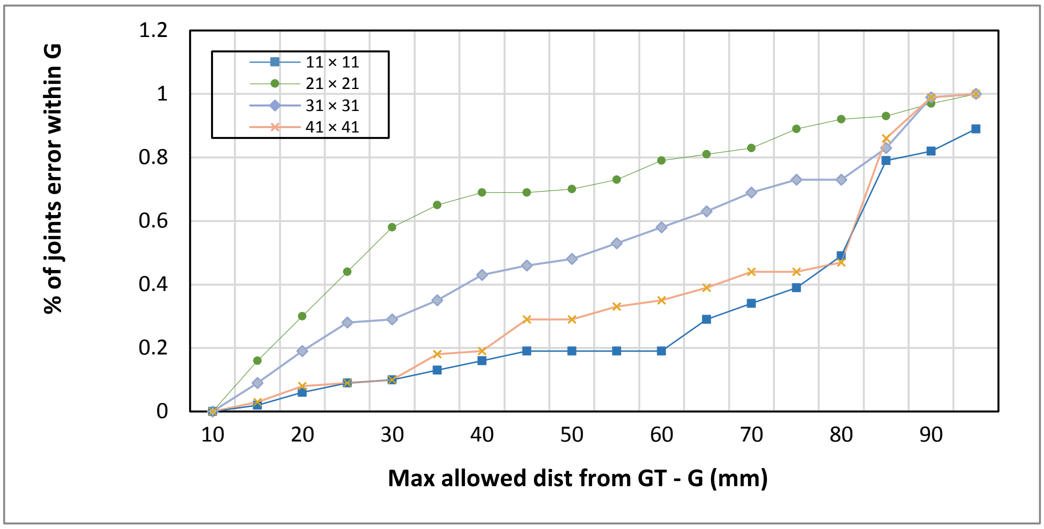


Figure 6.5 Evaluating the significance of the window size and the number of components used for spanning the prior pose space. The graph illustrates the percentage of joint pose prediction with a margin of error, G for different window sizes.

The same demarcation of the dataset was applied in training the pose-estimation network. Hence 10,000 pose-depth pairs are used in its training.

6.3.2 Baseline Comparison

To optimize the performance of the proposed technique two significant parameters were experimented with. Those include the window size (used in stereo matching) and the number of components used to store pose prior information. The window size determined the size of the input stereo pair regions that were fed into the similarity network for comparison and subsequently, the number of weights in the similarity network. From Figure 6.5, one can identify a gradual improvement in the accuracy as the size of the window reduces. 41×41, 31×31 and 21×21 window sizes yielded 18.23%, 35.54% and 65.22% of accurately predicted joint positions within an error of $35mm$, respectively. This trend stopped when a window size of 11×11 window was applied, resulting in 13% accurate predictions (see Figure 6.5). It can be speculated that superiority of 21×21 is due to the fact that at that window size, enough details (and visual cues) are present but also the size is not too large such that it allows for

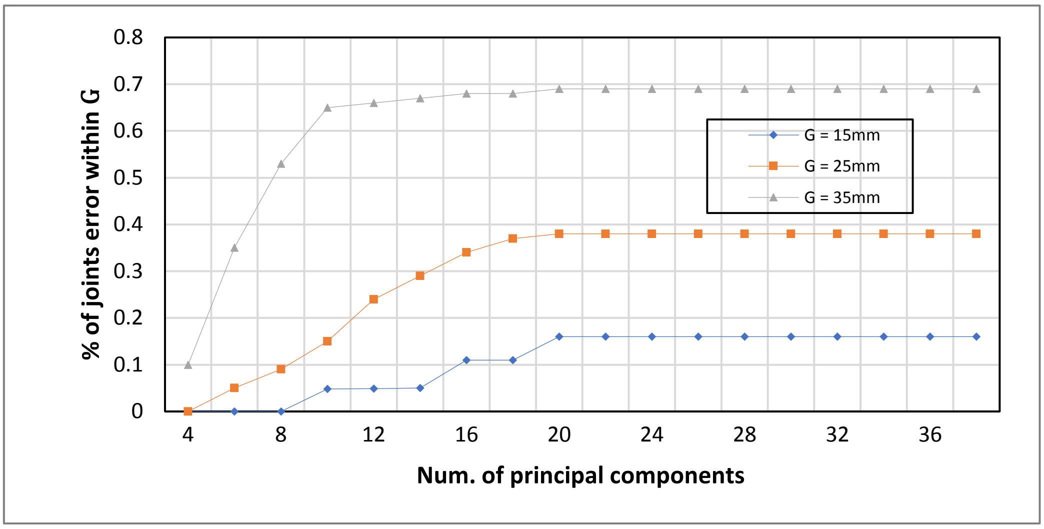


Figure 6.6 Evaluating the significance of the window size and the number of components used for spanning the prior pose space. The graph presents the percentage of correctly predicted joint position as a different number of components are used.

spurious matches to be made. A second parameter was the number of components used. Recall from Eq. 6.17 and 6.18 that from the $3 * K$ components only N are used. The significance of the number of components used is presented in Figure 6.6. This illustrates the increase in the percentage of accurate joint predictions as the number of components increases, however, this improvement in prediction performance stops after 10 to 18 of the most significant components have been used.

As well as the parameter evaluation, three baseline comparisons were made. The first was predicting the pose using a single shot depth estimation; the second was predicting pose without the pose prior and the third was constraining proposes hand poses.

Single-shot depth recovery: For a given stereo capture, all potential matching pixels are evaluated along the epipolar line on the corresponding stereo pair under the similarity network and a greedy search approach is applied to establish a disparity image. Next, the pose-estimation network is applied to directly estimate for the pose. This is often the prominent approach to depth-based pose estimation from a stereo framework, i.e. where depth is estimated in a "single-shot" of greedy search and then

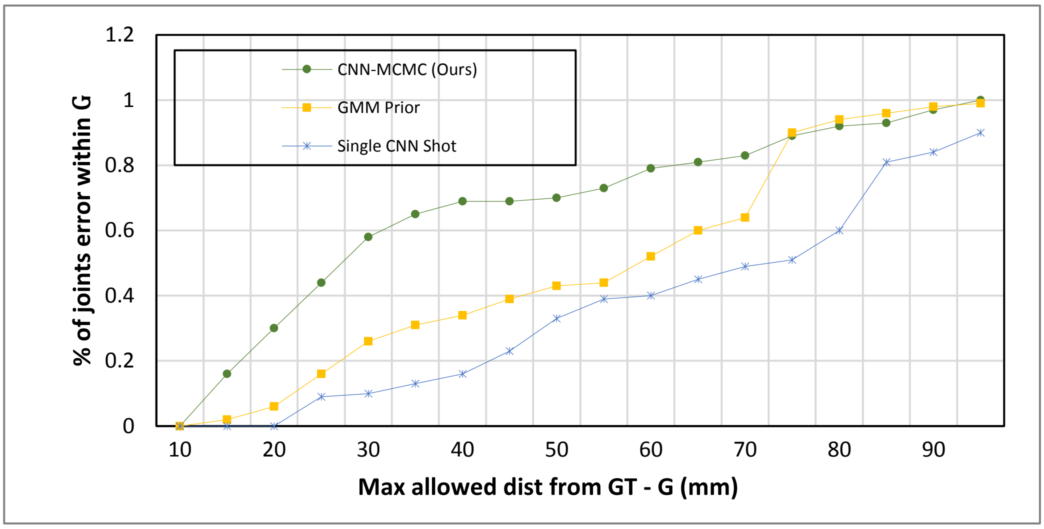


Figure 6.7 A baseline comparison of the proposed approach. The graph illustrates the percentage of accurately predicted joint pose prediction (within a margin of error), for the approach in comparison to the single shot depth estimation and to GMM based prior.

used to resolve for the pose as in [64]. Figure 6.7 validates the hypothesis presented in Section 6.1.1. The superiority of the jointly optimal, iterative depth proposal is apparent here, particularly at lower error thresholds. The ability to continuously re-evaluate the depth solution whilst resolving for pose contributes to this performance. In fact, there is a 389.8% increase incorrectly predicted joint positions (within a 35mm error margin) when the proposed approach is taken in comparison to the single shot approach. Although this superiority diminishes as the error threshold increases, the proposed iterative approach produces a more accurate hand pose estimation from stereo capture. The qualitative results in Figure 6.10 (4th row) corroborate this result, as better pose estimation is achieved with the proposed approach in comparison, particularly in the 1^{st}, 4^{th} and 5^{th} columns.

GMM prior: Another component of the framework is the pose prior. The effectiveness of the PCA based approach is evaluated by comparing it against a GMM (Gaussian Mixture Model) based approach. For this, expectation maximization is applied to establish a $3 * K$ dimensional GMM model that represents the probability of a pose

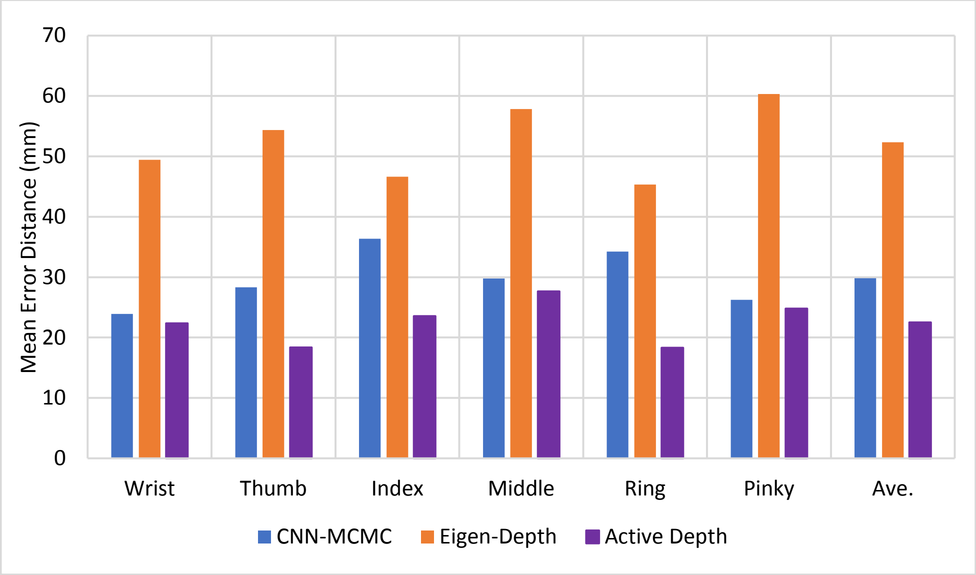


Figure 6.8 A baseline comparison of the proposed approach. Bar chart showing the mean joint position error per finger for the proposed approach, the work proposed in [2] and RGBD camera based pose estimation.

(as in [116]). Experiments were carried out to establish the optimal component. The performance of this approach is presented in Figure 6.7. Again, results show the superiority of the PCA based model, with the proposed approach producing a 109.6% increase incorrectly predicted joint positions (within a 35mm error margin). This is largely due to first identifying the highly discriminating components in the pose subspace before establishing a prior model. This superiority is replicated shown in Figure 6.10 (3rd row), particularly in the 1^{st}, 3^{rd} and $6th$ columns. The PCA based approach better constraints for a more realistic hand pose.

Although the superiority of the proposed approach against the baseline comparison is evident, there still exists some failure cases (see Figure 6.11). It can be observed that approach fails in scenarios were the hand is not properly outstretched. It also appears to suffer due to ambiguity between similar poses as illustrated in the first example, where it predicts that the pinky, ring and middle fingers to be outstretched when actually, it was the index, middle and ring finger that were outstretched.

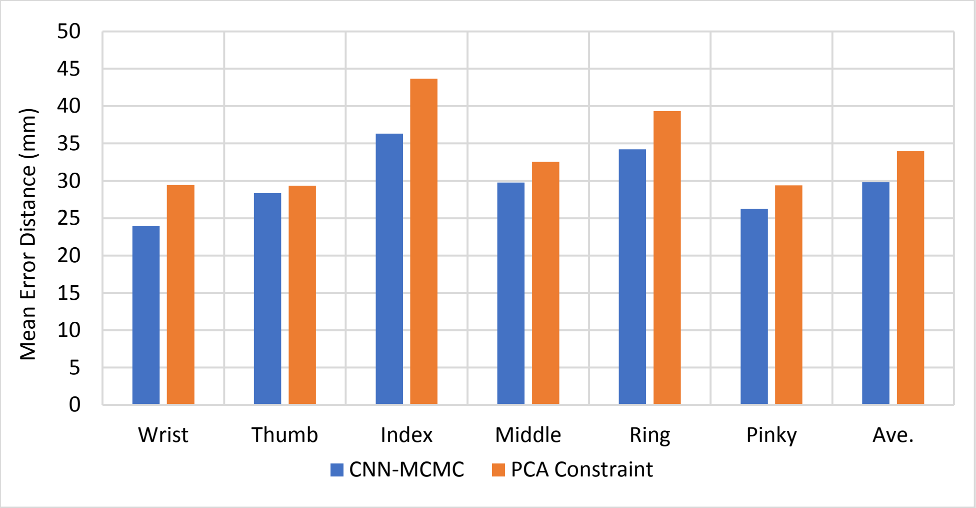


Figure 6.9 A baseline comparison of the proposed approach. Bar chart showing the mean joint position error per finger for the proposed approach, and using constrained poses.

Constrained pose: Recall that the pose prior is applied to penalize unrealistic poses (based on how a resolved pose deviates from the poses in the prior dataset). More specifically, using Eq. 6.17 and 6.18 the prior probability of a hand pose that was resolved in a discriminative manner (using the pose estimation network) was used to penalize a proposed depth. A more common application of PCA is to strictly constrain hand pose solution to belong to the range of the matrix $\boldsymbol{E}$ i.e. that pose is strictly a linear combination of the principal components, $[\boldsymbol{e}_1, ..., \boldsymbol{e}_N]$. This alternative was investigated in this subsection. As a result, instead of choosing the pose solution from the set $\{\boldsymbol{h'}_{\{i\}}\}$ (where $\boldsymbol{h'}_i$ is the resolved pose from the i^{th} depth proposal using the pose estimation network), it was chosen from $\{\boldsymbol{h}^*_{\{i\}}\}$ such that

$$\boldsymbol{h}^* = \boldsymbol{E}^+\boldsymbol{\alpha}^* + \boldsymbol{\mu}. \tag{6.23}$$

Consequently, the potential pose solution is constrained to be a linear combination of the principle component. This is different to the approach of the framework presented above, where the optimum pose solution is chosen strictly from the resolved pose (using the pose estimation network), however potential poses are penalized based on the strength of their affinity with the closest linear combination of the principle component. The result of this baseline comparison is presented in Figure 6.9. The inferiority of this approach is immediately apparent, yielding a 13.75% increase in the average error. A possible conjecture for this is that the potential poses are less dependent on the observed stereo capture than on the information in the prior dataset. As a result, although a more realistic pose solution is produced, this conforms less to the observed pose instance.

6.3.3 Comparison Against ELNF

For further evaluation, the performance of the presented work is validated against the previous ELNF framework. Recall that the ELNF framework regresses for robust hand depth estimation using an Eigen leaf node based variant of a regression forest. Chapter 4 motivates the approach with depth recovery specifically for hand pose estimation. To evaluate this, the pose-estimation network was applied to directly regress for pose from the recovered depth using the approach in [2]. The performance is presented in Figure 6.7. Again, like the single-shot method, this approach performs significantly worse than the newly proposed join optimization approach. On average this MCMC based approach preforms 29.55% better than the ELNF approach in Chapter 4 (29.80mm to 42.32mm error). This corroborates the significance of jointly optimizing for both pose and depth. The single shot approach assumes a high-quality depth prediction and will yield a poor result when the preceding depth estimation is poor.

6.3.4 Comparison Against RGBD Sensors

To evaluate the significance of the MCMC based approach in the general context of gesture recognition, the accuracy of the pose estimation prediction made is compared to pose estimations made from depth images to that acquired from the RGBD camera. Again pose estimation is computed using the pose-estimation network. Figure 6.8 presents the evaluative comparison. Compared to the MCMC based approach, the

RGBD based pose prediction was more accurate in predicting thumb, the index and ring finger joints. This is due to large variance in their 3D position across the training and testing dataset. Across all five fingers, the mean joint position error of estimated pose from the RGBD depth image is 21.99mm, this is only 9.304mm lower than the mean joint position error of the proposed technique (30.802mm). Considering the low-quality nature of the stereo camera used, the proposed approach exhibits robustness against inconsistency and noise in stereo capture to an extent that it is on par with pose estimation made from an active depth sensor. This is significant, as it shows a

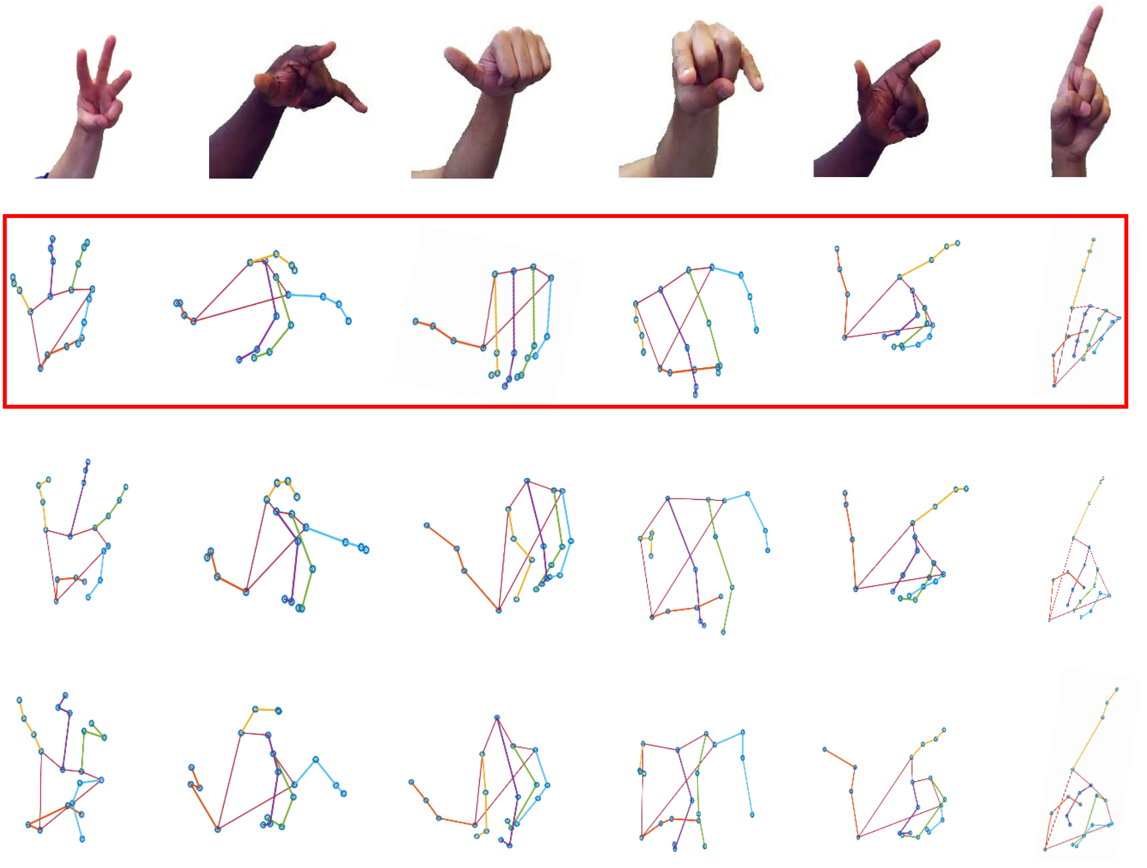


Figure 6.10 Qualitative results of pose estimation using real stereo captured poses. The reference image of the stereo pair is shown in the 1^{st} row. The results from the proposed method are presented in the 2^{nd} row. The 3^{rd} row shows the pose estimation result from using the method but with a GMM pose prior while 4^{th} row shows result from using the single-shot CNN.

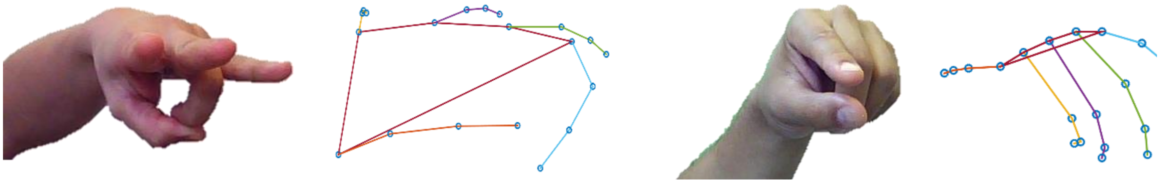


Figure 6.11 Examples of failure cases. Observe that the approach fails in scenarios were the hand is not properly outstretched. It also appears to suffer due to ambiguity between similar poses as illustrated in the first example, where it predicts that the pinky, ring and middle fingers to be outstretched when actually, it was the index, middle and ring finger that were outstretched.

potential of overcoming the drawbacks in RGBD, discussed in the introductory chapter, without a significant drop in the accuracy of pose estimation.

6.3.5 Summary

In this chapter, a novel approach to pose estimation from stereo capture is presented which proposes an MCMC-CNN approach to joint optimization. The presented approach stochastically proposes depth images with the aim of ensuring that this proposed depth is consistent with the observed stereo capture and that it resolves to a realistic hand pose. The inferred pose from the optimum depth is identified as the pose solution to the stereo input. It has been shown experimentally that this joint optimization approach outperforms the conventional single shot depth estimation approach. This is largely owing to the fact that with the sequential approach, the pose estimation (from depth) part of this framework will need to be robust against initially regressed depth that may be erroneous.

A negative impact of this sequential framework is the slow computation time. This is significant has it precludes any real-time application. An interesting improvement on this will be a closed-form solution to the estimation of the depth configuration by establishing a parametric relationship between the depth configuration and the stereo cost. The idea here is to first establish a matching cost value for all potentially matching pixels to a given superpixel of interest. Consider a superpixel, x_j, with a centroid pixel, $\boldsymbol{x}_j$, and a set of potentially matching pixels $P = \{\boldsymbol{x}_j^k\}_{k=1}^K$ in the corresponding image. Note that the pixel points in set P are contiguously located and that there exists a corresponding matching cost $c_j^k = F(\boldsymbol{x}_j^k)$, where the domain of the function $F()$ is discrete. Given the current framework presented in Chapter 6, $F()$ is modelled by the similarity network. Now, assume a scenario where the input domain is continuous (instead of discrete pixel locations) then the function $F()$ (i.e. the similarity network) can be modelled with a mixture of Gaussians. This entails initially computing the cost of all potentially matching pixels (conventionally using the similarity network) to get $\{(\boldsymbol{x}_j^1, F(\boldsymbol{x}_j^1)), ..., (\boldsymbol{x}_j^K, F(\boldsymbol{x}_j^K))\}$ and then applying maximum likelihood to compute $\arg\max_{\boldsymbol{\theta}} Pr\Bigg(\{(\boldsymbol{x}_j^1, F(\boldsymbol{x}_j^1)), ..., (\boldsymbol{x}_j^K, F(\boldsymbol{x}_j^K))\}|\boldsymbol{\theta}\Bigg)$, where $\boldsymbol{\theta}$ is a set of Gaussians' parameters. Consequently, $F()$ can be approximated to an analytical function as a sum of Gaussians, as in $F(\boldsymbol{x}) \approx G_{\mu_1,\sigma_1}(\boldsymbol{x}) + ... + G_{\mu_N,\sigma_N}(\boldsymbol{x})$, where there are N Gaussian components and $G_{\mu,\sigma}()$ is the conventional Gaussian function of mean, μ and standard deviation, σ. This will allow for parallelizing the CNN execution in a single run for an improved runtime. Given the Gaussian-based modelling of $F()$, samples of depth solutions can be generated simultaneously drawn and evaluated in parallel, consequently overcoming the sequential bottleneck of MCMC sampling.

This chapter presents a joint optimization approach that is robust against potential error in the depth estimation pre-step. Thus, there is not burden on the pose estimation framework to be robust against erroneous depth recovery. A consequence of the proposed approach is that it iteratively revise for errors in depth proposal. This allows for simultaneous correction of proposed depth estimation and the resulting pose estimation to jointly optimize the likelihood of the depth and hand pose estimation given the stereo input. The approach proposed is a semi-generative approach that is experimentally proven to work on different sizes and tones of hand without pre-calibration. This is significant as it shows the potential of estimating hand pose articulation based on stereo input with accuracy that is on par with using commercially available RGBD sensors.

This concludes the work done in the thesis. As previously outlined, the following chapter is an experimental one, where the performance across all the frameworks presented in the last three chapters is compared.

Chapter 7

Sequential vs Joint Optimization

This chapter compares the two approaches to stereo based pose recovery proposed in the thesis. Recall that the initial approach in the first two frameworks introduced (Chapters 4 and 5) was to first solve for depth and then recover pose from the estimated depth. The second approach (presented in Chapter 6) jointly optimizes for depth and pose. The results in Chapter 6 show the superiority of the pose recovery via joint optimization compared to sequential (depth is first solved before pose) optimization. However, it is not clear that the improvement in performance is solely or at least mainly due to multiple depth proposals rather the matching criteria used (i.e. the similarity network). To this end, this chapter presents additional experiments to assess the effect of the generative depth proposal of Chapter 6 compared to discriminative depth estimation.

7.1 Significance of multiple depth proposal

To investigate the significance of the multiple depth proposals, modifications were made to the first two proposed frameworks: (ELNF and CRRF). The aim here is to first investigate if these depth estimation frameworks could be used in a generative manner (like the approach presented in Chapter 6). Secondly to find out how it compares to the CNN based matching cost. To achieve this, rather than use the regressed depth recovered via using ELNF or CRRF, the probabilistic output from both were used. Dataset C was used in the following experiments.

To address the first question, recall that in the ELNF framework, the optimal depth value is identified as $\arg\max_d Pr(d|f_{\boldsymbol{\theta}})$ and then used for pose estimation. An alternative to this would be to propose different depth maps based on the probability $Pr(d|f_{\boldsymbol{\theta}})$. The same can be applied in the case of CRRF, again the output posterior over depth levels are used to propose different depth solutions. These proposed depth solutions can, in turn, be used to regress for pose using the pose-estimation network (Chapter 6), before applying the pose prior based on the linear constraint (see Section 6.1.4).

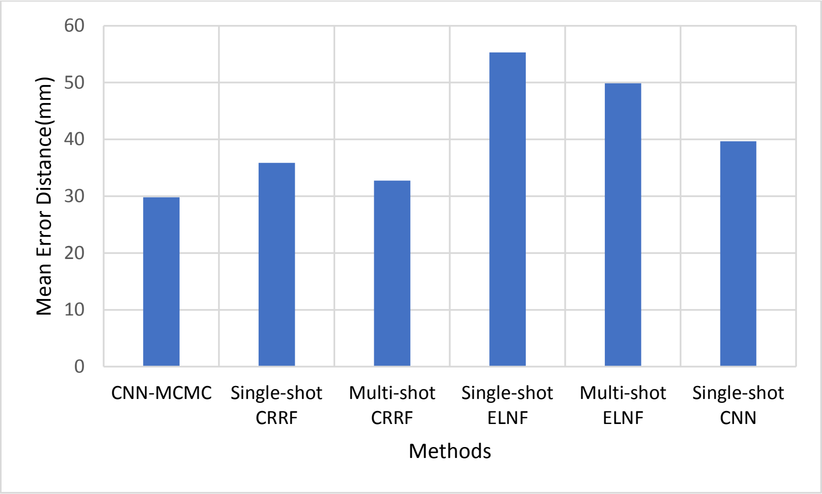


Figure 7.1 An illustration of performance of ELNF, CRRF and MCMC-CNN based on the Mean Error Distance. Note the superiority in the performance of the CRRF and MCMC-based techniques in comparison to the ELNF-based approach. Also, the impact of the multi-shot approaches (in comparison to the single-shot approaches)can also be observed.

This is the approach taken to investigate the significance of depth proposal. Two sets of baseline comparisons were made: a single-shot approach and a multi-shot approach. Here single-shot indicates the recovery of a single depth map and then regressing for depth whilst multi-shot entails joint optimization for depth and pose using multiple

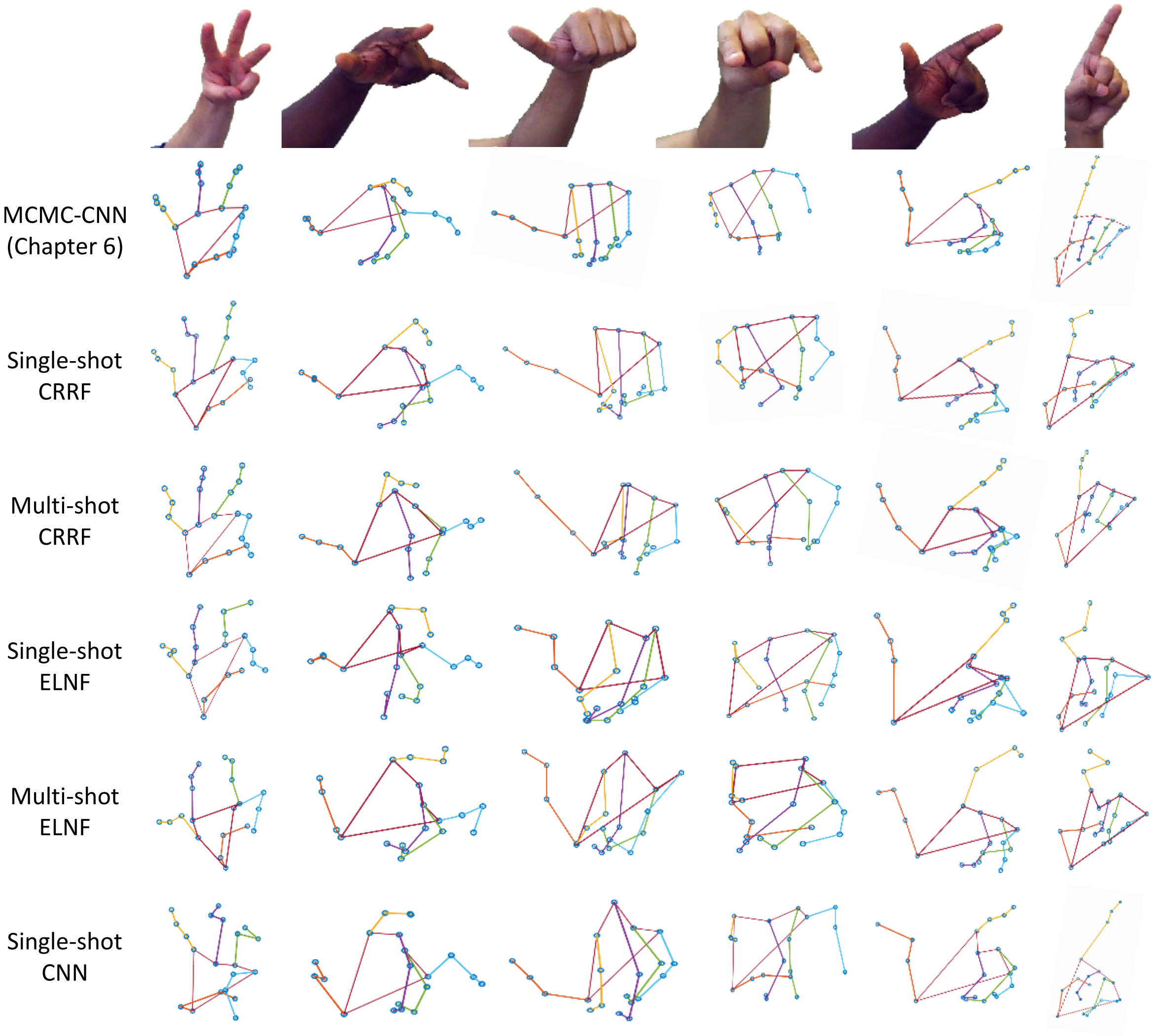


Figure 7.2 Qualitative results of pose estimation using real stereo captured poses. The reference image of the stereo pair is shown in the 1^{st} row. The results from the MCMC-CNN (Chapter 6) approach are presented in the 2^{nd} row. The 3^{rd} and 5^{th} rows present pose estimation results from using single-shot CRRF and ELNF approaches respectively. The 4^{th} and 6^{th} rows present pose estimation results from using multi-shot CRRF and ELNF approaches respectively. The final row shows result from using the single-shot CNN.

generated depth maps. These results are presented in Figures 7.1 (quantitatively) and 7.2 (qualitatively).

Single-shot approach: The first comparison that was made was across single-shot baselines. These included three approaches, namely: single-shot ELNF, single-shot CRRF and single-shot CNN. Single-shot ELNF entails using the optimal depth prediction from the ELNF framework to regress for pose. Similarly, single-shot CRRF and single-shot CNN baselines use depth recovered using the CRRF framework and the similarity network (introduced in Section 6.2) respectively, to regress for hand pose. Note that all pose estimation from depth was carried out using the pose estimation network (Section 6.2). Observe from Figure 7.1 that of all the three single shot based approaches, the CRRF depth based pose estimation produces the best pose accuracy, reducing the mean error of ELNF based baseline by 35.17% and CNN (similarity network) based baseline by 9.53%. This clearly shows the superiority of the depth resolved by the CRRF framework against both the similarity network and the ELNF approach. Relative performances are also reflected in the qualitative results presented in Figure 7.2.

Multi-shot approach: The second set of baseline comparisons made was the use of multiple proposals using the depth posteriors from both ELNF and CRRF; and comparing against the MCMC-CNN approach. The aim here is to investigate to what extent the multiple-depth-proposal aspect of the MCMC-CNN approach improves its performance. An immediate observation is that all the multi-shot approaches improve on the single shot counterpart both on average and individually. Mean error of CRRF, ELNF and CNN based pose estimation is reduced by 8.57%, 10.9%, and 25.65% respectively. From the qualitative perspective, a more stable quality in the pose estimations are also observed when comparing the multi-shot approaches (2^{nd}, 4^{th} and 6^{th} rows) against the single-shot ones (3^{rd}, 5^{th} and 7^{th} rows). A more interesting observation is that, even though the CRRF based pose estimation performed better than the CNN (similarity network) approach under a single shot approach, the opposite is observed when under a multi-shot approach. Qualitatively, whilst there exist instances where the single-shot approach produces more realistic and accurate pose estimation than the multi-shot approach (e.g. 6th column of the 3rd and 4th row in Figure 7.2), on aggregate the multi-shot produced better poses. It was hypothesised that this has to do with the fact that MCMC sampler in the MCMC-CNN framework samples from the joint distribution of the depth and joints (see Eq. 6.20 from Section 6.1.5) whilst the multi-shot CRRF approach samples solely based on depth probability distribution.

Another key observation is the effect of the PCA-based pose prior. It is noticeable that some of the predicted hand poses are often not realistic. This is because the pose prior cost of the resolved pose (from a proposed depth configuration) is one of the costs for evaluating the proposed depth and the corresponding pose. Hence the inclusion of the pose prior does not restrict the possible pose predictions to realistic poses but instead penalizes pose prediction that does not conform to realistic poses. This is an important distinction and hence explains the possibility of some of the baseline comparisons above predicting non-realistic hand poses (as seen in Figure 7.2).

7.2 Summary

This chapter concludes the work and experiments conducted for this thesis. The chapter investigates the significance of the different features proposed. More specifically it investigates the comparative performance of six approaches to hand pose estimation from stereo capture. These approaches are mainly grouped into single-shot approaches and multi-shot approaches.

Experimental results indicate the superiority of joint optimization, as the multi-shot approaches perform better than the single-shot approaches. This is due to the fact that it allows for optimization to be done jointly in the depth domain, as well as, on the pose domain. However, under a single-shot framework, CRRF out-performs the deep learning approach (Siamese network) as a hand depth estimation approach for pose recovery. To investigate the possibility of performing joint optimization using CRRF (as opposed to Siamese network), the probability based depth prediction of the CRRF technique (see Eq. 5.16), is used to propose depth solutions (similar to MCMC in the MCMC-CNN framework) by sampling from the posterior distribution, $\boldsymbol{y}^*$.

These proposed depth solutions are used to resolve for pose (using the pose-estimation network) before being evaluated, using the pose prior model (Section 6.1.4) and the initial depth probability. Although this yielded a slightly improved result on the single-shot CRRF approach, it does not perform as well as the MCMC-CNN approach. A possible reason for this is that the MCMC sampler in the MCMC-CNN framework samples from the joint distribution of the depth and joints whilst the multi-shot CRRF approach samples solely based on depth probability distribution.

Chapter 8

Conclusion

This thesis has explored the use of passive vision for the estimation of hand pose using a Stereovision system composed of adjacent RGB cameras. Such a camera rig does not project light into the scene and therefore has complementary advantages to depth imaging, including less energy consumption. However, hand pose estimation in this context is a more challenging computer vision problem, one that has received less attention in the literature. The thesis has addressed this gap by proposing a number of novel frameworks that contribute to the solution of pose recovery from stereo capture. This includes an innovative application of the regression forest technique for upgrading disparity to depth, proposing a cost function that estimates using ELNF that is more suitable for regression. Improving upon this by using a CRF-Random Forest framework for predicting superpixel based depth whilst simultaneously constraining for smoother depth prediction. Finally, a framework that combines jointly optimal depth and hand pose estimation in a unified framework using Markov-chain Monte Carlo (MCMC) sampling and deep learning. This research is motivated by the possibility of estimating articulation with the input of stereo cameras from an egocentric, stereoscopic perspective. The work is inspired by the human vision, which can efficiently discern articulations and perform tracking activities with passive, binocular input.

The thesis was motivated with four primary objectives. The first was "to propose, develop, implement and evaluate hand depth estimation framework". The work presented in Chapter 4 and 5 meet this objective. Specifically, the CRRF approach (Chapter 5) presents a robust implementation of this objective by introducing a highly accurate hand depth estimation framework. This second objective was "to propose, develop and implement a framework for hand pose/articulation estimation from recovered hand depth". This was addressed to a lesser extent by the thesis. This is largely due to the fact that there exist several robust techniques that readily addresses hand pose recovery from depth in literature. Nonetheless, the thesis presents the implementation of the pose estimation framework in Chapter 6. Here a deep learning model (based on the work of Oberweger et al. in [5])was implemented for recovering hand pose from depth. Consequently, a novel solution was not proposed here, instead, the implemented approach was based on prior literature. The third objective was "to propose a new approach for joint stereo reconstruction and pose estimation of a hand from stereo inputs". This objective was met in Chapter 6 with the development of the MCMC-CNN framework, which estimates hand pose articulation from stereo capture using MCMC-based proposal on two deep network model. Again, this was met to a great degree with pose estimation on per results from a depth-based input. Lastly, the fourth objective was "to evaluate the performance of the stereo-based pose estimation approach to RGBD input-based approach". This was achieved to some extent. The experiments presented in Chapter 6 made a comparison between hand pose estimation based on RGBD input and stereo input. However, this is quite limited. This is owed to the fact that the ground truths used in this thesis are based on RGBD depth sensors, hence it will be ill-posed to compare the prediction from stereo input-based approach (that was trained with RGBD-based ground truth) to RGBD-based approach. A more thorough comparison will be to train the stereo-based approach on a third-party ground truth such as laser-based depth measurements.

8.1 Summary of the work presented

The first chapter introduces the problem of pose estimation from stereo capture. Specifically, this chapter highlights the key challenges that make the problem unique. These include the fact that disparity estimation is a useful step to inferring shape information for robust pose estimation. However, disparity recovery from the hand is a slightly more challenging task in contrast to a scene containing arbitrary objects. Largely due to the large texture-less region but also the fact that the relatively cheap stereo camera used in this thesis means there are lots of inconsistencies between both stereo images.

In Chapter 2, a brief introduction to the key concepts in the thesis is presented. This includes the pinhole model, multi-view geometry, camera calibration, homogeneous coordinates, machine learning frameworks (such as Random Forest, CNN, CRF etc.) as well as an introduction to the datasets used in the thesis. In Chapter 3, a more comprehensive outline of the current literature concerning the related work was given. This was largely categorized into three topics, namely: stereo algorithms; hand pose estimation; and hand pose recovery from stereo capture. The limited volume of work done specifically on stereo based hand pose recovery was made apparent.

The next three chapters examined the main work completed. In Chapter 4, a novel variant of the Regressive Random Forest framework is presented, and it is noted that the technique is applicable beyond the context of this research. The significance of this approach has been conveyed experimentally. The qualitative and qualitative experiments show the capacity of ELNF and how it allows for higher accuracy, even at low tree depth, owing to the implicit reduction of entropy as a result of marginalizing based on pixel features. In the fifth chapter, an alternative data-driven method to estimate an accurate depth map of a hand from a stereoscopic camera input is proposed by introducing a superpixel-based regression framework that takes advantage of the smoothness of the hand's depth surface. To this end, a novel method that combines a closed-form Conditional Random Field with learned weights and a Regressive Random Forest (RRF) with adaptively selected expert trees is presented. This is to model the mapping from a stereo RGB image pair to a depth image. The intuition behind the RRF is that it adaptively selects different stereo-matching measures as it implicitly determines matching pixels in a coarse-to-fine manner. While the RRF makes depth prediction for each superpixel independently, the CRF (Conditional Random Field) unifies the prediction of depth by modelling pair-wise interactions between adjacent superpixels. This was the work done on hand depth estimation for pose recovery. The work in these two chapters addresses the first research questions: "How can high-quality depth information be recovered from stereo capture?"; and "How should texture-less hand regions and radiometric differences in cameras be addressed?". A high confidence depth estimation was achieved my using a machine learning based approach to establishing the matching criteria. Experiments showed that rather than having a single matching criterion, the quality depth images can be computed by having multiple matching criteria and then applying the machine learning tool to learn how to combine these different matching criteria. This is particularly the case with the proposed CRRF model. Also, matching costs that have been experimentally tested to be robust against radiometric differences in stereo camera pairs were used.

The last two research questions were addressed in the sixth chapter. These were: "How can hand pose be estimated from stereo image capture?"; and "How does this compare to depth based estimation?". In this chapter, a different approach to the problem of pose estimation from stereo capture was taken. Here, rather than solve for depth and then for pose in a sequential manner, an alternative approach is presented. Potential depth solutions are stochastically proposed and then evaluated in two paths. First by evaluating how consistent such depth is with the observed stereo capture and secondly using the proposed depth to resolve for pose and evaluating how realistic the resolved pose is. Experiments show that the performance of pose estimation from stereo capture is still on par with depth-based pose estimation.

In Chapter 7, an evaluation of the significance of the depth proposal is evaluated. Specifically, the depth estimation frameworks proposed in Chapter 4 and 5 are used to propose depth in a similar way to the approach in Chapter 6.

8.2 Limitations of the Work

The ELNF framework (Chapter 4) works well on a constrained set of poses (i.e. only frontal planar poses) however when a more arbitrary hand orientation is present in the dataset (see Section 5.3), this approach performs significantly less well. Another, limitation in the work is that the techniques still require a disparity estimation and hence the performance of the framework relies on the consistency in stereo-matching pre-step.

As discussed above, the CRRF framework presented in Chapter 5 yields a robust depth estimation of hand pose, however there some significant limitations. The most obvious of these is the need for a skin segmentation step that precedes the stereo-matching algorithm. Whilst this does not affect the performance of the technique itself it will affect the shape of the recovered hand depth. False hand segmentation could be an issue in scenarios where the recovered depth is to be used as a feature for further analysis. For instance, in [20, 59, 60] the feature for pose estimation from a depth image is dependent on the shape of the hand. Hence, the trivial foreground estimation that can be done on RGBD input (by ignoring pixels with depth values larger than a threshold) cannot be exploited here. However, egocentric based approach to hand region segmentation, like [117], could be a potential solution to this. This could be used as a pre-step to the approach presented in this thesis. Another potential limitation of this technique is that it quantizes the depth space, limiting the depth sensing reach or resolution. Whilst larger depth sensing reach can be learned by adapting the training set appropriately, this will lead to a computation cost vs. depth reach/resolution trade-off. Since larger depth reach or resolution will require more depth levels (and hence increase in the size of the matrix $\boldsymbol{B}$ and $\boldsymbol{Y}$), the computational expenses of the technique increases (see Figure 8.1). A solution to this problem might be to use a logarithmic scale for depth so that less resolution will be given to depth prediction far away (which is often more significant) and vice versa.

The MCMC based approach in Chapter 6 also has some limitations. As well as the hand region segmentation limitation (similar to the case in CRRF model), another

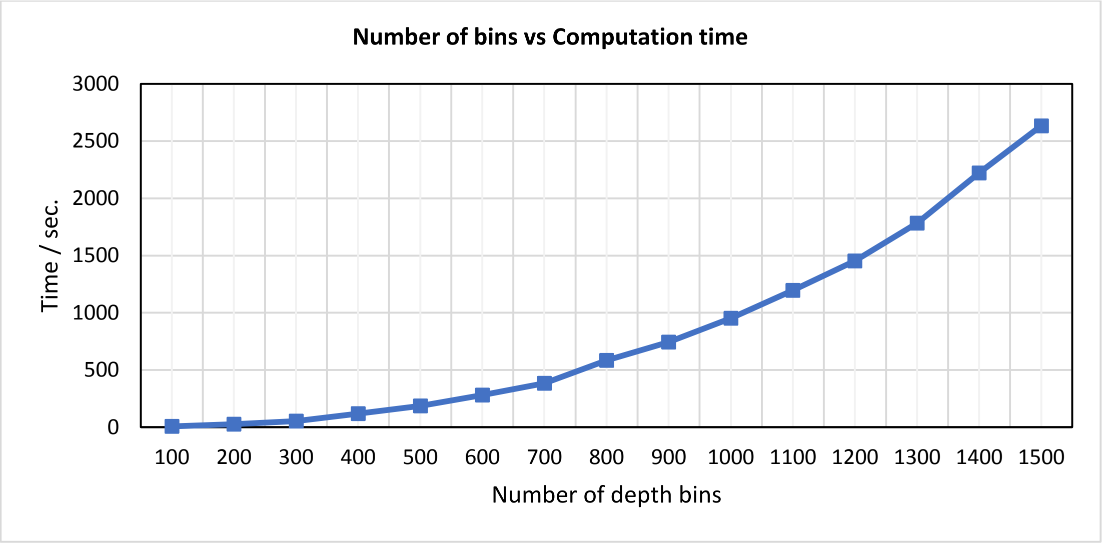


Figure 8.1 Graph showing the effect of the number of depth bins on the execution time of the CRRF technique.

drawback is the fact that it requires an iterative approach in order to generate/propose a depth solution. This is a limitation because it hinders depth proposals to be evaluated in a parallel manner. Since the proposal of a new potential depth solution is dependent on the evaluation score of the prior proposals. Also, the methods in the thesis only work on a single hand. Whilst a trivial solution could be to introduce a hand detection framework, that initializes different execution instances of the frameworks (presented in the thesis) to run on each hand detected, this does not address the problem of self-occlusion of the hands. The occlusion that can occur from the interaction of multiple hands in the scene or even interaction with other arbitrary objects is not accounted for in the frameworks presented.

Another limitation of the work is the broadness of the poses, the number of participants and the general size of the dataset used. In order to cover a large variety of hand poses more data instances will be needed, particularly in the context of deep learning based approach. A more robust output would have been achieved by using substantially larger dataset (in the order of 100, 000) that consists of a larger number of hand poses and participants. Lastly, the approaches presented do not run in real-time, with is essential to human-computer interaction. The complexity of the framework causes a larger execution time.

8.3 Practical Application

Although the objective of this thesis is to explore the possibility of implementing hand pose estimation based on stereo input and not to necessarily to produce a real-time algorithm, it is useful to discuss the computaional performance of the work in this thesis for completeness. The need for much greater computational efficiency is clear if anything approaching real-time tracking is to be achieved. Some potential avenues to improving the proposed technique have been discussed in the previous chapters. For instance, in the ELNF framework, the use of superpixels as opposed to predicting depth for each pixel was discussed, this combined with a GPU based parallel prediction of

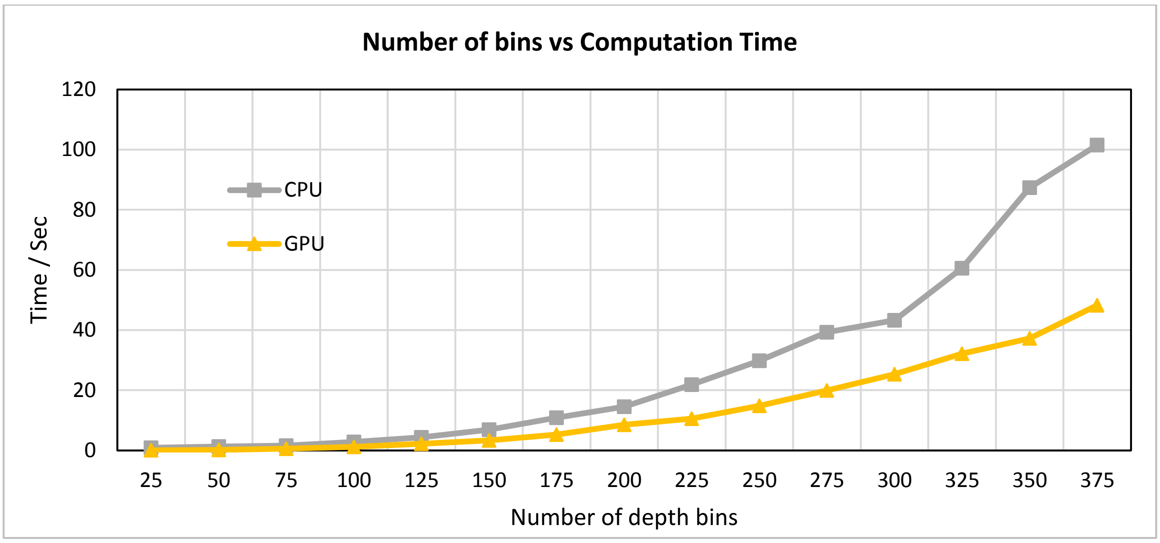


Figure 8.2 Graph showing the effect of the number of depth bins on the execution time of the CRRF technique before and after GPU optimization.

depth at each pixel (similar to the work of Shotton et al. in [65]) can yield a real-time implementation.

As discussed previously, a significant bottleneck in the run-time of the CRRF model is the inversion of $\boldsymbol{B}$. It was suggested that an improvement on the execution time of this framework is parallel computation. An example of this is demonstrated here. Using the CUDA[1] parallel computing platform, a dense matrix inversion routine (cublasSgetrfBatched) from the CULA[2] high-level linear algebra library was used to optimize the matrix inversion on an NVIDIA Titan X GPU with 12GB memory. The result of this experiment is presented in Figure 8.2. This resulted in a 52.45% decrease in computation time. It is worth noting, that future developments of both hardware and software may mean even faster performance, and by sacrificing depth bins there is hope that real-time performance may be possible in the future.

Lastly, with the MCMC-CNN approach, it was discussed that the sequential nature of the framework could be improved by the parallel proposal and hence evaluation of depth solution. The idea here is to pre-compute the matching cost value for all potentially matching pixels to a given superpixel of interest. Then solving analytically for the optimum set of matching pixels with the lowest re-projection error that yields hand poses that best conform to the prior dataset.

8.4 Future Work

This section outlines potential directions for future work.

Generalizing CRRF and MCMC-CNN framework: A stance that has been maintained throughout this work is that the frameworks proposed in this thesis are applicable to other machine learning problems. Hence a plausible research direction for the future will be to explore the performance of these frameworks on other problems. For instance the application of the CRRF framework to predicting the behaviours of spatially related entities. A typical example of this could be in predicting the product

[1] https://developer.nvidia.com/cuda-zone

[2] The CUBLAS library contains routines for batched matrix factorization and inversion.

consumption of a population based on their geographical relationship. Similar to the neighbouring superpixels of hands having similar depth, it can be assumed that people living in adjacent neighbourhoods will have similar needs etc. The same applies to the MCMC-CNN approach. An interesting application that can be explored is human action detection. Here a joint optimization of detecting humans in the scene as well as determining what action the person is doing can be addressed with a variant of the MCMC-CNN approach.

Addressing egocentric perspective: Another potential research direction is the consideration of the pose estimation problem from an egocentric perspective. This will introduce some inherent problems, like the rapid movement of the camera etc. However, the poses are more constrained, which could be exploited, for instance in the context of the pose prior.

Temporal tracking: The work presented in this thesis largely explores static hand poses. Though this is a common approach in the literature [65], it does not explore the possibility of utilizing the temporal information in the data. The fact that a pose was observed (predicted) in the prior frame can inform the pose that can be expected in the next frame. Of course this in itself is not a novel concept, in fact, this is the basis of the generative model based pose estimation. However, an interesting avenue to explore would be how to embed this temporal knowledge into the existing framework. A possible approach could be to augment the hand pose prior, $Pr(\boldsymbol{H})$, in the MCMC-CNN framework to include a conditional prior i.e. $Pr(\boldsymbol{H}_t|\boldsymbol{H}_{t-1})$, where t is the index of the frames.

Hybrid tracking: Another potential avenue in future research is to explore the possibility of a hybrid system that combines stereo and RGBD-based inputs to resolve and track hand poses. This will be a useful application of the work in this thesis, by supplementing the non real-time performance of the stereo-based frameworks presented in this thesis with RGBD inputs. Conversely, information from stereo-based input can be used to supplement the diminished performance of RGBD-based inputs in scenarios such as outdoor usage, close range etc. The idea here is that stereo-based approach (such as MCMC-CNN) that makes predictions in non real-time intervals can be used to re-initialized both measure depth image and resolved posed of a real-time RGBD system to yield a far more robust approach.

Bibliography

[1] Hands on with microsoft hololens. http://uk.pcmag.com/consumer-electronics-reviews-ratings/39112/feature/hands-on-with-microsoft-hololens. Accessed: 2018-1-28.

[2] 8 real-world uses for microsoft hololens. https://www.makeuseof.com/tag/8-real-world-uses-microsoft-hololens/. Accessed: 2018-1-28.

[3] The industrial applications of hololens. https://www.pegusapps.com/en/insights/the-industrial-applications-of-hololens. Accessed: 2018-1-28.

[4] 6 unique uses for microsoft's hololens. https://www.pcmag.com/feature/335878/6-unique-uses-for-microsoft-s-hololens/1. Accessed: 2018-1-28.

[5] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, Santiago, Chile, December 11th 2015.

[6] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, 2013.

[7] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, Boston, Massachusetts, United States, June 7th 2015.

[8] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, Columbus, Ohio, United States, June 17th 2014.

[9] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169, 2014.

[10] Emad Barsoum. Articulated hand pose estimation review. *arXiv preprint arXiv:1604.06195*, 2016.

[11] Minoru 3d webcam to launch at ces. https://newatlas.com/minoru-3d-webcam/10660/. Accessed: 2017-10-08.

[12] 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*, 33(9):1793–1805, 2011.

[13] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 3–19, Dublin, Ireland, June 26th 2000.

[14] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 101.1 – 101.11, Dundee, Scotland, August 29th 2011.

[15] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2095, Barcelona, Spain, November 6th 2011.

[16] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Dart: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, 39(3):239–258, 2015.

[17] Matthieu Bray, Esther Koller-Meier, and Luc Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, 106(1):116–129, 2007.

[18] Javier Romero, Hedvig Kjellström, and Danica Kragic. Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 458–463, Anchorage, Alaska, United States, May 8th 2010.

[19] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3325–3333, Santiago, Chile, December 11th 2015.

[20] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 852–863, Florence, Italy, October 7th 2012.

[21] Paschalis Panteleris, Nikolaos Kyriazis, and Antonis A Argyros. 3d tracking of human hands in interaction with unknown objects. In *Proceedings of the British Machine Vision Conference*, pages 123.1–123.12, Swansea, United Kingdom, September 7th 2015.

[22] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–528, New York City, New York, United States, June 17th 2006.

[23] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[24] Rostam Affendi Hamzah and Haidi Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016, 2015.

[25] Heiko Hirschmuller and Daniel Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599, 2009.

[26] John P Lewis. Fast normalized cross-correlation. In *Vision interface*, volume 10, pages 120–123, Québec City, Canada, May 16th 1995.

[27] HP Moravec. Toward automatic visual obstacle avoidance. 5th int. joint conf. *Artificial Intelligence*, page 584, 1977.

[28] Uwe Stilla Franz Rottensteiner, Helmut Mayer Boris Jutzi, and Matthias Butenuth. Photogrammetric image analysis. 2011.

[29] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 151–158, Stockholm, Sweden, May 2nd 1994.

[30] Nicolas Gac, Stéphane Mancini, Michel Desvignes, and Dominique Houzet. High speed 3d tomography on cpu, gpu, and fpga. *EURASIP Journal on Embedded systems*, 2008:5, 2008.

[31] Ge Zhao, Ying Kui Du, and Yan Dong Tang. A new extension of the rank transform for stereo matching. In *Advanced Engineering Forum*, volume 2, pages 182–187, 2012.

[32] Li Ma, Jingjiao Li, Ji Ma, and Hanyue Zhang. A modified census transform based on the neighborhood information for stereo matching algorithm. In *Proceedings of the IEEE International Conference on Image and Graphics*, pages 533–538, Melbourne, Australia, September 15th 2013.

[33] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, Minnesota, United States.

[34] Geoffrey Egnal. Mutual information as a stereo correspondence measure. *Technical Reports (CIS)*, page 113, 2000.

[35] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[36] Heiko Hirschmüller, Peter R Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, 2002.

[37] Jiangbo Lu, Gauthier Lafruit, and Francky Catthoor. Anisotropic local high-confidence voting for accurate stereo correspondence. In *Proceedings of SPIE 6812, Image Processing: Algorithms and Systems VI*, pages 68120J–68120J, 2008.

[38] Jinglin Zhang, Jean-Francois Nezan, Maxime Pelcat, and Jean-Gabriel Cousin. Real-time gpu-based local stereo matching method. In *Proceedings of the Conference on Design and Architectures for Signal and Image Processing*, pages 209–214, Cagliari, Italy.

[39] Cevahir Cigla and A Aydın Alatan. Information permeability for stereo matching. *Signal Processing: Image Communication*, 28(9):1072–1088, 2013.

[40] Ke Zhang, Jiangbo Lu, Qiong Yang, Gauthier Lafruit, Rudy Lauwereins, and Luc Van Gool. Real-time and accurate stereo: A scalable approach with bitwise fast voting on cuda. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(7):867–878, 2011.

[41] Zucheul Lee, Jason Juang, and Truong Q Nguyen. Local disparity estimation with three-moded cross census and advanced support weight. *IEEE Transactions on Multimedia*, 15(8):1855–1864, 2013.

[42] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[43] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 467–475, Stockholm, Sweden, July 30th 1999.

[44] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewénius, and David Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504, 2009.

[45] Ying Wu, John Lin, and Thomas S Huang. Analyzing and capturing articulated hand motion in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1910–1922, 2005.

[46] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics (TOG)*, 28(3):63, 2009.

[47] TH Taehee Lee and Tobias Hollerer. Handy ar: Markerless inspection of augmented reality objects using fingertip tracking. In *Proceedings of the IEEE International Symposium on Wearable Computers*, pages 83–90, Boston, Massachusetts, United States, October 11th 2007.

[48] Martin de La Gorce and Nikos Paragios. A variational approach to monocular hand-pose estimation. *Computer Vision and Image Understanding*, 114(3):363–372, 2010.

[49] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143, 2016.

[50] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 3633–3642, New York City, New York, United States, April 18th 2015.

[51] Iason Oikonomidis, Manolis IA Lourakis, and Antonis A Argyros. Evolutionary quasi-random search for hand articulations tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3422–3429, Columbus, Ohio, United States, June 2014.

[52] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface*, pages 63–70, Regina, Saskatchewan, Canada, May 31st 2013.

[53] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[54] James S Supancic, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1868–1876, Sydney, Australia, December 03rd 2015.

[55] James John Kuch. *Vision-based hand modeling and gesture recognition for human computer interaction.* PhD thesis, University of Illinois at Urbana-Champaign, 1994.

[56] Kinect for xbox one. http://www.xbox.com/en-GB/xbox-one/accessories/kinect. Accessed: 2017-09-08.

[57] Simon J. D. Prince. *Computer Vision: Models Learning and Inference.* Cambridge University Press, 2012.

[58] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3224–3231, Sydney, Australia, April 4th 2013.

[59] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[60] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013.

[61] Bjoern Stenger, Paulo RS Mendonça, and Roberto Cipolla. Model-based 3d tracking of an articulated hand. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 310–315, Kauai, Hawaii, United States, December 11th 2001.

[62] Ying Wu, John Y Lin, and Thomas S Huang. Capturing natural hand articulation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 426–432, Vancouver, Canada, July 9th 2001.

[63] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Tracking the articulated motion of two strongly interacting hands. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1862–1869, Providence, Rhode Island, United States, June 18th 2012.

[64] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

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

[66] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, Columbus, Ohio, United States, June 17th 2014.

[67] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2456–2463, Sydney, Australia, December 3rd 2013.

[68] Javier Romero, Hedvig Kjellström, and Danica Kragic. Monocular real-time 3d articulated hand pose estimation. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pages 87–92, Paris, France, December 7th 2009.

[69] Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *Proceedings of the International Conference on 3D Vision*, volume 1, pages 319–326, Tokyo, Japan, December 8th 2014.

[70] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, Boston, Massachusetts, United States, June 7th 2015.

[71] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 644–651, Columbus, Ohio, United States, June 17th 2014.

[72] Muhammad Asad, Enguerrand Gentet, Rilwan Remilekun Basaru, and Greg Slabaugh. Generating a 3d hand model from frontal color and range scans. In *In the Proceedings of the IEEE International Conference on Image Processing*, pages 4589–4593, Quebec City, Canada, September 27th 2015.

[73] Fred L Bookstein Principal Warps. Thin-plate splines and the decompositions of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6), 1989.

[74] Sean Ryan Fanello, Cem Keskin, Shahram Izadi, Pushmeet Kohli, David Kim, David Sweeney, Antonio Criminisi, Jamie Shotton, Sing Bing Kang, and Tim Paek. Learning to be a depth camera for close-range human capture and interaction. *ACM Transactions on Graphics (TOG)*, 33(4):86, 2014.

[75] 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, Sydney, Australia, December 3rd 2013.

[76] Vincent Lepetit, Pascal Lagger, and Pascal Fua. Randomized trees for real-time keypoint recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 775–781, San Diego, California, United States, June 20th 2005.

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

[78] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, San Francisco, California, United States, June 13th 2010.

[79] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, Columbus, Ohio, United States, June 17th 2014.

[80] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, Boston, Massachusetts, United States, June 7th 2015.

[81] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Proceedings of the Neural Information Processing Systems Conference*, pages 1799–1807, Montreal, Canada, December 13th 2014.

[82] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3593–3601, Nevada, United States, June 26th 2016.

[83] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[84] Georg Poier, Konstantinos Roditakis, Samuel Schulter, Damien Michel, Horst Bischof, and Antonis A Argyros. Hybrid one-shot 3d hand pose estimation by exploiting uncertainties. *arXiv preprint arXiv:1510.08039*, 2015.

[85] Paschalis Panteleris and Antonis Argyros. Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo. *arXiv preprint arXiv:1705.05301*, 2017.

[86] Javier Romero, Danica Kragic, Ville Kyrki, and Antonis Argyros. Dynamic time warping for binocular hand tracking and reconstruction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2289–2294, Pasadena, California, United States, May 19th 2008.

[87] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 432–439, Madison, Wisconsin, United States, June 18th 2003.

[88] Iason Oikonomidis, Nikolaos Kyriazis, Konstantinos Tzevanidis, and Antonis A Argyros. Tracking hand articulations: relying on 3d visual hulls versus relying on multiple 2d cues. In *In the Proceedings of the International Symposium on Ubiquitous Virtual Reality*, pages 7–10, KAIST, Daejoen, South Korea, July 10th 2013.

[89] Romer Rosales, Vassilis Athitsos, Leonid Sigal, and Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 378–385, Vancouver, Canada, July 9th 2001.

[90] Nikolaos Kyriazis and Antonis Argyros. Scalable 3d tracking of multiple interacting objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3430–3437, Columbus, Ohio, United States, June 17th 2014.

[91] Grégory Rogez, James S Supancic, and Deva Ramanan. First-person pose recognition using egocentric workspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4325–4333, Boston, Massachusetts, United States, June 7th 2015.

[92] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 294–310, Amsterdam, Netherlands, October 8th 2016.

[93] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. *Computer Vision–ECCV 2012*, pages 640–653, 2012.

[94] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[95] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[96] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, 2008.

[97] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees.* Chemical Rubber Company Press, 1984.

[98] Networkx. https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.dag.is_aperiodic.html. Accessed: 2017-09-08.

[99] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, Kerkyra, Greece, September 22nd 1999.

[100] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–154, 2005.

[101] Mokhtar M Hasan and Pramod K Mishra. Superior skin color model using multiple of gaussian mixture model. *British Journal of Science*, 6(1):1–14, 2012.

[102] Mokhtar M Hasan and Pramod K Mishra. Novel algorithm for skin color based segmentation using mixture of gmms. *Signal & Image Processing*, 4(4):139, 2013.

[103] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 666–673, Kerkyra, Greece, September 22nd 1999.

[104] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis.* Springer, 2013.

[105] Tarn Duong et al. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 21(7):1–16, 2007.

[106] Rigged hand - blend swap. https://www.blendswap.com/blends/view/66039. Accessed: 2018-02-20.

[107] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[108] Rilwan Remilekun Basaru, Chris Child, Eduardo Alonso, and Greg Slabaugh. Quantized census for stereoscopic image matching. In *Proceedings of the International Conference on 3D Vision*, volume 2, pages 22–29, Tokyo, Japan, December 8th 2014.

[109] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, Boston, Massachusetts, United States, June 7th 2015.

[110] Frank J Aherne, Neil A Thacker, and Peter I Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4):363–368, 1998.

[111] Pietikäinen Matti, Hadid Abdenour, G Zhao, and T Ahonen. *Computer Vision Using Local Binary Patterns*. Springer, Dordrecht, 2011.

[112] Middlebury website. http://vision.middlebury.edu/stereo/data/. Accessed: 2017-09-08.

[113] Robert T Collins and Peter Carr. Hybrid stochastic/deterministic optimization for tracking sports players and pedestrians. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 298–313, Zurich, Switzerland, September 12th 2014.

[114] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.

[115] Vlfeat matconvnet. http://www.vlfeat.org/matconvnet/. Accessed: 2017-09-08.

[116] Michael Burke and Joan Lasenby. Single camera pose estimation using bayesian filtering and kinect motion priors. *arXiv preprint arXiv:1405.5047*, 2014.

[117] Cheng Li and Kris M Kitani. Pixel-level hand detection in ego-centric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3570–3577, Portland, Oregon, United States, June 23rd 2013.

[118] Stereoscopy website. http://www.stereoscopy.com/faq/aerial.html. Accessed: 2014-06-19.

[119] Michael Oren and Shree K Nayar. Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14(3):227–251, 1995.

Appendix A

A.1 Quantized Census

Stereo-matching for disparity recovery has been used in a wide range of applications, including object recognition, object tracking, robotic navigation and even recovery of landscape topography from aerial photography [118]. This broadness of scope implies that a good correspondence matching system has an inherent need to be adaptive to different illumination conditions. Basic stereo-matching systems utilise a simple matching cost function to identify corresponding points in images taken from multiple perspectives (often two) with the assumption of identical intensity level at points of corresponding image locations. This is will be referred to as the *consistency assumption*. As a result of different illumination conditions, amongst other factors, the consistency assumption rarely holds and more complex cost functions are required to account for radiometric differences.

As mentioned above, several conditions breach the consistency assumption. The illuminating conditions are a major issue as they can seldom be controlled. This is as a result of non-Lambertian surfaces and specular reflection [119]. The difference in illumination to the light sensor component of the cameras will result in the same point in 3D space being perceived at different intensity levels. Another cause of radiometric differences is the inconsistency of the image capturing devices themselves. Properties such a salt and pepper noise, Gaussian noise, vignetting, gain setting (linear and non-linear) etc. will generally be inconsistent in multiple devices hence resulting in radiometric differences.

The above discussion, establishes that the requirement for robustness against radiometric differences is essential for a stereo-matching system to be used in real application like hand pose estimation from stereo capture. To this end an improved variant of the Census cost [29] function is proposed.

A.1.1 Census-Hamming Distance

The Census cost function is implemented as a non-parametric local transformation to the window of interest whilst the Hamming Distance is the similarity measure that utilizes the result of this transformation. Consider a local neighborhood N_p with a center pixel p and intensity $I(\boldsymbol{p})$. Assuming a rectified stereo pair, with a pixel, p in the left image, the corresponding pixel in the right image is $\boldsymbol{p} - \boldsymbol{d}$. For a single image (left or right) the intensity level of the center pixel is compared to that of surrounding pixels (denoted with q) within the considered neighborhood to generate "a bit string representing the set of neighboring pixels whose intensity is less than" or greater than $I(p)$ [29]. Formally,

$$I_c(\boldsymbol{p}) = \mathbb{I}[I(\boldsymbol{q}) > I(\boldsymbol{p})]. \tag{A.1}$$

The binary result from Eq. A.1 is concatenated across all the pixels in the neighborhood. The Hamming distance between the transformed neighborhoods in both corresponding images is then computed. This is the number of bit-positions that are different in two bit strings. The larger this value the more dissimilar the two neighborhoods in question. Whilst the Census-Hamming combination is a strong cost function against some radiometric changes, it has one major flaw in that it is not invariant to non-monotonic radiometric distortions. Consider a $1D$, image region with 5 pixels shown in Figure 3a.

53	99	100	102	135

(a)

53	101	100	99	135

(b)

Figure 3 Intensity levels of 1D Image region before (a) and after (b) non-monotonic distortion

Here the Census transform for this region would be: [0011]. If the image is distorted non-monotonically, the relative ordering of intensity level is lost, for example, as shown in Figure 3b. This would result in a different Census transform of $[0,1,0,1]$. Even though the distortion was slight, this results in a 50% error.

A.1.2 Quantized Census (QC)

Quantized Census is proposed in an attempt to compensate for the deficiency in the Census matching cost. QC applies a less rigid system that accommodates for non-monotonic distortions to the ordered level of intensity. Just like in the Census case, QC utilizes the comparative intensity of the middle pixel and the neighboring pixels, but is also sensitive to intensity gradient. It transforms the intensity level at each pixel within the neighborhood of interest to a *quantized equivalent* of the difference in the intensity value of the middle pixel to that of the surrounding ones. This does not only provide information on the order of relative intensity but also, to some extent the magnitude. Continuing with our previous notation, the transformation is as follows:

$$D(\boldsymbol{q}) = Q_N\{I(\boldsymbol{q}) - I(\boldsymbol{p}\}. \tag{A.2}$$

where $Q_N\{\}$ denotes N bins of quantization. It is important to note that the subtraction operation that precedes the quantization could yield values that range from negative to positive. Hence quantization is applied in the range of -255 and +255 (intensity color range). For example if 16 bins were used, then the quantized value would range from -7 to 0 in the negative range and 0 to 7 in the positive domain. The effect of this equation is that subtle non-monotonic distortions, that do not preserve the order of pixel intensity, will not be detected by the cost function. This is significant as imaging devices would not perfectly capture subtle intensity changes in a scene.

Looking at the plot one can partially identify where some of the pairs of corresponding points are. However, it is also notable that the relative ordering of intensity is not consistent especially in the low textured region. This will also be the case in the

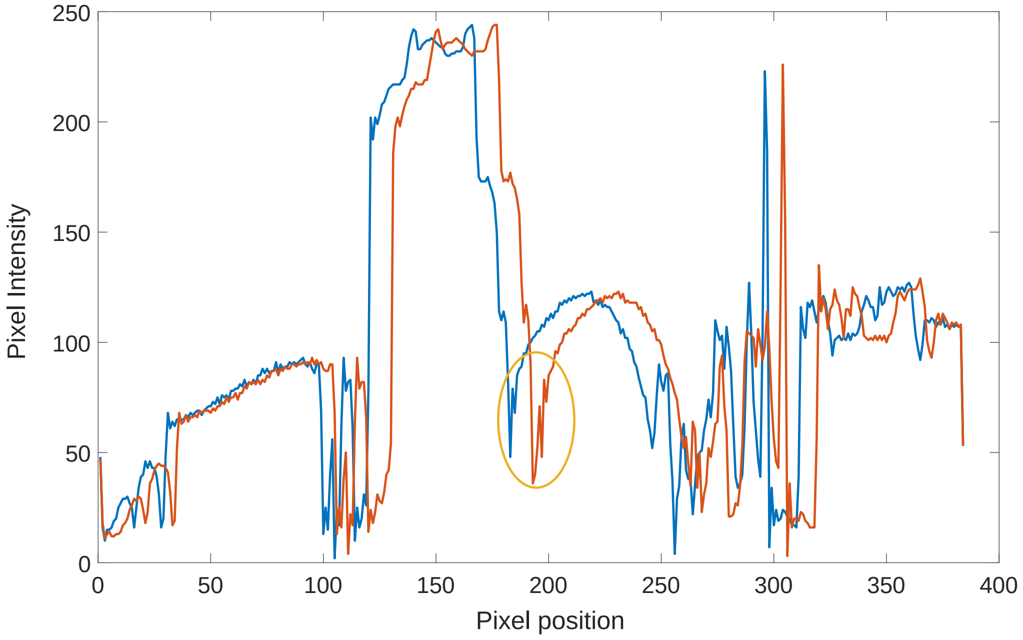


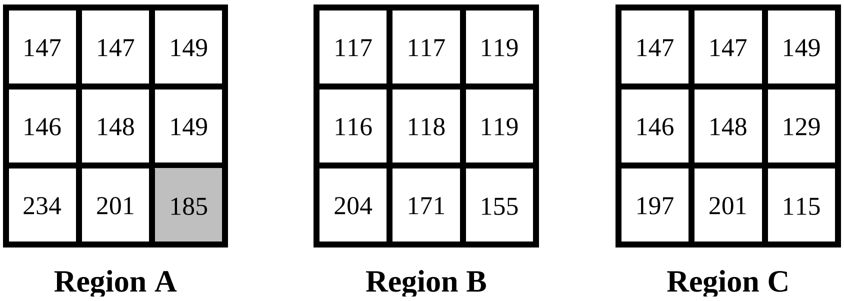
Figure 4 1D intensity row plot of the pair of the Tsukuba stereo image from the Middlebury dataset [112]. The red plot indicates the intensity along a horizontal line on the right image whilst the blue plot indicates the intensity along the left image. The yellow marker indicates regions of inconsistency in the relative ordering of intensity between the left and right image.

presence of distortions like Gaussian noise. Subtle intensity distortion due to Gaussian noise with low signal to noise ratio would be ignored as a result of quantization. For example, looking back at Figure 3, if a quantized difference (with 16 bins) is applied then the resulting transform for both region A and B will be $[-1, 0, 0, 1]$. Hence, it permits for the subtle non-monotonic distortion.

Whilst the modification in Eq. A.2 has improved robustness, it immediately poses a problem. A key strength of the Census cost function is that it is robust to distortions like salt and pepper noise. It achieves this by not using an aggregative costing technique (in terms of intensity levels) like in SAD or NCC. Each erroneous pixel contributes equally to the cost, making it insensitive to outliers. With our modification, the intuitive cost would have been to acquire the sum of absolute or square differences. Of course this would be to the detriment of how well the cost function performs against outliers. This is because outliers that instigate huge quantized difference would influence the sum of absolute difference. Taking inspiration from the work of Fischler and Bolles, RANSAC algorithm [95], the number of outliers were used as opposed to summing the cost at each pixel. This has made the cost function invariant to radiometric changes that do not preserve the relative ordering of pixel values. Formally, the Quantized Census stereo-matching cost is defined as

$$C_{QC}(\boldsymbol{p}, \boldsymbol{d}) = \sum_{\boldsymbol{q} \in N_{\boldsymbol{p}}} \mathbb{I}[|D_L(\boldsymbol{q}) - D_R(\boldsymbol{q} - \boldsymbol{d})| < T], \tag{A.3}$$

where T is a threshold value. This cost is applied to the transformed neighborhood pair that is tested for correspondence. Here D_L and D_R refers to the quantized pixel differences acquired by Eq. A.3. To illustrate how Eq. A.3 tolerates salt and pepper noise, consider a 3-by-3 region in a first image (region A in Figure 5), and two potentially matching 3-by-3 regions in a second image (regions B and C in Figure 5. These regions have been chosen to illustrate a linear gain scenario where the ground truth matching region is in fact region B with a bias of 30.



147	147	149
146	148	149
234	201	185

Region A

117	117	119
116	118	119
204	171	155

Region B

147	147	149
146	148	129
197	201	115

Region C

Figure 5 Intensity value of neighborhood.

The resulting transformation (with 32 bins of quantization) for regions A, B and C will be.

0	0	0
0		0
5	3	2

Transformed Region A

0	0	0
0		0
5	3	2

Transformed Region B

0	0	0
0		-1
3	3	-2

Transformed Region C

Figure 6 Transformed values of neighborhood using Eq. A.2.

First, note the invariance of the cost function to radiometric differences, while the relative pixel values are preserved. Next, assume that the shaded pixel in region A is a randomly altered pixel value as a result of noise. Figure 7 illustrates the effect of the intensity level on the cost function had the sum of been used instead of a threshold (as in Eq. A.3). A significant degree of distortion in a single pixel is enough to affect the result of the cost function. If the intensity was distorted to less than 185, region C would wrongly be chosen as the best match. Instead, by considering the number of pixels pairs with absolute differences less than a particular threshold, this is rectified. In the above scenario, regardless of the intensity of pixel X, the number of outliers will be the same.

More generally, for a region in the first image and another potentially matching region in the second, the difference between the intensity of the middle pixel and that

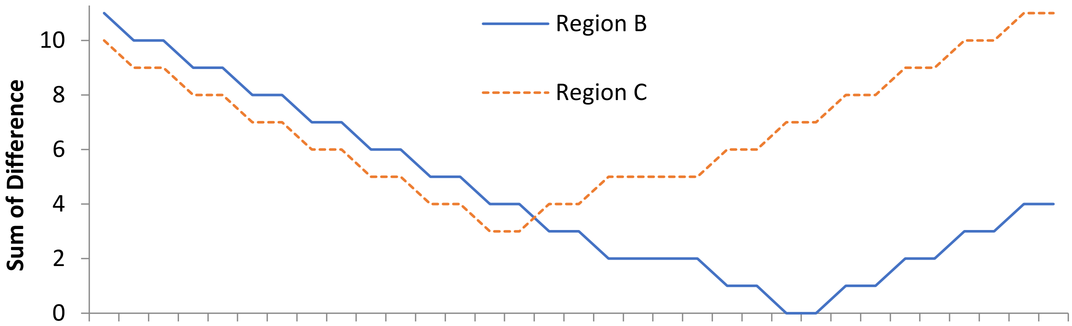


Figure 7 Resulting Cost when sum of the absolute difference of the transformed regions is used to compare Region A to Region B and C.

of neighborhood pixels are acquired respectively. These differences are then quantized into an experimentally determined number of bins. The absolute difference of the both transformed region is taking and the result is compare to a threshold (that is also experimentally determined) to generate a binary region. The sum of the binary region is to be maximized across all potentially matching regions. Further details on experiments as well as results can be seen in [108].