Semi Automatic Hand Pose Annotation using a Single Depth Camera

Thesis Submitted in Fulfillment of the Requirements for the Degree of PhD in Computer Science

by

Marnim Galib

1001427030

Supervised by: Prof. Vassilis Athitsos

Department of Computer Science and Engineering

UNIVERSITY OF TEXAS AT ARLINGTON

August 2022

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Vassilis Athitsos, for his continuous support and patience during my PhD study. I would also like to thank my academic committee members Dr. Dajiang Zhu, Dr. Christopher Conly, Dr. Farhad Kamangar, Dr. Won Hwa Kim, and Dr. Alex Dilhoff for their interest, assistance and valuable feedback. I am also thankful to my labmates in the VLM Lab@UTA for their help, cooperation and knowledge sharing.

Finally, this journey wouldn't be possible without the unconditional love and support of my parents, Md. Mizanur Rahman, late Monoara Jesmin and my grandparents, aunts and uncles. I am thankful to my wife, Nowrosh Islam for her love and encouragement and my brother, Mahir Mubassir Sadik for his enthusiasm in my work.

Abstract

This thesis investigates the problem of 3D hand pose annotation using a single depth camera. While hand pose annotations are critically important for training deep neural networks, creating such reliable training data is challenging and manual labor intensive. Current datasets that rely on manual annotation on real images are limited in size due to the difficulty of annotating them. Although, large datasets have been generated using tracking based methods followed by manual refinement, these methods are prone to annotation errors due to tracking failure. Synthetic images have also been used to create large datasets but synthetic frames does not capture the sensor characteristics such as noise while also producing kinematically implausible and unnatural hand poses. We propose a semi-automatic method for efficiently and accurately labeling the 3D hand key-points in a hand depth video. The process starts by selecting a subset of frames that are representative of all the frames in the dataset and the user only provides an estimate of the 2D hand key-points in these selected frames. We use this information to infer the 3D location of the joints for all the frames by enforcing appearance, temporal and distance constraints. Finally, we demonstrate that our method can generate 3D training data more accurately using less manual intervention and offering more flexibility in comparison to other state-of-the-art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the introduction of RGB-D sensors, the landscape of computer vision has changed in the last decade. By combining an RGB camera with a depth camera, it provides not only the color and light intensity, but also the distance of each observed pixel. There are great expectations that these sensors will lead to a boost of new 3D perception-based applications in the fields of robotics and visual & augmented reality. Integration of RGB-D sensors in smartphones and tablets promises to make these applications even more popular.

Hand motion capture with RGB-D sensor gained a lot of research attention lately as hands are the primary body part to interact with the surrounding environment. As we adopt smartphones, smartwatches and smartglasses, traditional input devices such as keyboard, mouse are not an option anymore. On-screen interaction is hard due to small size of display. If hand poses can be inferred correctly, hand interaction with the surrounding environment can be used as input for different applications. Therefore, markerless capture of human hands is a very interesting problem and we focus our attention on the improvement of current hand pose estimation methods.

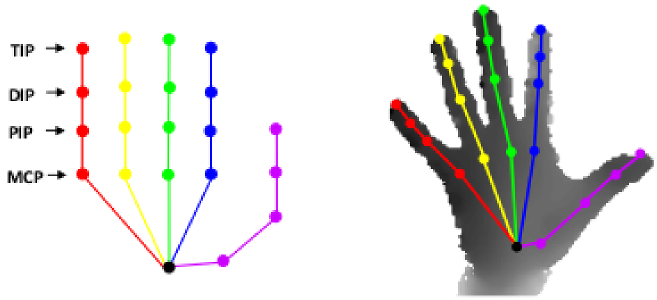

FIGURE 1.1: The Human Hand Model using 21 keypoints [1]

1.2 The Human Hand

The human hand has a complex structure with many individual degrees of freedom. Some researchers have used 16 joints to model the human hand, while others have used 21 joints. In the following figure, we show the more common approach of modeling a human hand following [1] that uses 21 keypoint locations and 31 degrees of freedom. Each finger consists of four keypoints : the metacarpophalangeal (MCP), proximal interphalangeal (PIP), distal interphalangeal (DIP) joints and we additionally consider the fingertip as another keypoint (TIP) . Each joint is assigned a specific number of Degrees of Freedom (DoF), depending on the physiology of the joint. The wrist has six DoF as it can move freely around the body comprising 3D translation and global rotation. The MCP joints have two DoF , PIP, DIP and TIP points have one DoF . This sums up to a total of 31 DoF. These DoFs are controlled by 38 muscles in the hand and forearm allowing the hand to articulate the hand bones in a coordinated manner. The DoFs of the hand cannot all be independently controlled. In spite of this limitation, hands are capable of dexterous movements such as touching, grasping, gesturing and other manipulations.

1.3 Problem Statement

Hand pose estimation is the task of finding the hand keypoints to infer the pose information from a given image or video frame. Vision based hand pose estimation is a challenging field where we estimate the position of hand joints without using any specialized sensing equipment but a camera. Early works used 2D images which makes this problem very difficult due to ambiguities in the 2D images. With the advent of

3D sensors, such as structured-light or time-of-flight sensors, inferring 3D joint locations became much more feasible. More recently, hand pose estimation from color images have become possible by using large amounts of training data from multiple viewpoints [12]. Other devices such as Leap Motion sensors have also been used for hand pose estimation, however they still lack robustness and accuracy for real world applications.

In this work, we focus on 3D hand pose annotation using a semi-automatic approach, which aims at minimizing the manual labor intensive hand pose annotation while still providing reasonably accurate annotations. We use markerless pose estimation, which implies that we use bare hands as input without any markers such as bands or colors. In a more formal definition, we aim at mapping an input depth image D, to the corresponding 3D hand pose P. This mapping task is not significantly different from other pose estimation problems such as body pose. However, hand pose estimation has unique challenges because of the complex physiology of the human hand.

1.4 Challenges

Markerless 3D hand pose estimation is a difficult task that involves dealing with several challenges. Although some of these challenges are not unique to hand pose estimation, often times hand pose estimation poses a more difficult configuration of the body pose estimation task. Some of the many different challenges are mentioned here briefly:

Occlusions : While for rigid objects the location and pose can be inferred from non-occluded parts, that is not possible for highly articulated human hands. As mentioned earlier, human hands have a complex anatomy with many DoF that can move independently. Hand pose annotation becomes more challenging in egocentric views or when two hands are interacting with each other, but for this work we are not focusing on either of these scenarios. Even with manual annotations, it is sometimes difficult to figure out the correct keypoint locations due to occlusions.

Size and Shape : Humans hands are of different shapes and sizes. Not only the size of a human's hand changes from childhood to adulthood, there are significant difference between adults as well. In the field of computer vision, related works have not focused too much on these variations.

Noisy Data : 3D Hand pose estimation requires robustness to noisy data. The noise may come from camera sensors such as missing regions or having noisy outliers in the depth images. In RGB images, there can be shadows and different lighting conditions that can introduce noise. Also, the annotations are sometimes noisy or erroneous especially in large datasets because of the automatic nature of annotating them.

Self-similarity of fingers : Hand pose estimation poses a different problem in the aspect that unlike body pose estimation hands contain self-similar parts that are hard to distinguish. This obviously results in estimation errors especially when one or more fingers are occluded.

Fast Motion : Hands can move fast, and individual fingers can move quickly irrespective of wrist movements. Human hand can perform actions very fast and can cause motion blur as a result. This is especially challenging in tracking based methods that use the pose of the previous frame as initialization.

Detection and Segmentation : In the hand pose annotation task, detection and segmentation of hands is assumed as a prerequisite using depth thresholding or using color. However, in real-world environments, this can be a non-trivial task. For example, depth thresholding fails if the hand is close to the body or another object nearby and color thresholding may fail when a similar color is present in the background. Background clutter and lighting conditions may also pose challenges for accurate hand detection.

1.5 Dissertation Organization :

We start with a brief introduction of the hand pose estimation problem. In Chapter 2, we review some of the methods and datasets that are being used for hand pose estimation. We provide a detailed analysis of existing methods for creating a large scale dataset and define our scope of work. We also provide a brief overview of our proposed semi-automatic pipeline for hand pose estimation. In Chapter 3, we introduce our proposed method with necessary details for each step of the pipeline. We show our preliminary results and define the research plan in Chapter 4. Finally, we conclude our discussion with future directions in the last chapter.

1.6 Outlines of Proposed Work :

For this work, we choose depth camera based real hand pose annotation. Recent real hand pose datasets that are annotated manually [15, 16] are limited to a few thousand frames. Although, large real hand pose datasets [6, 14, 17] exist, these datasets are prone to annotation errors due to tracking failure. We, therefore, propose a semi-automatic pipeline that produce highly accurate hand pose annotation within a very small error margin of manually annotated images.

Firstly, given a hand depth video, we select a small number of frames that are representative of all the frames in the depth sequence. The frames will be selected automatically using a distance function and we refer these representative frames as reference frames. We then annotate the 2D hand joint locations in these representative frames and using this manual annotations we infer the 3D location of all the frames by enforcing appearance, distance and temporal constraints. We propose that by annotating a small fraction of the frames (5-10 %) we can generate 3D annotations more accurately than other state-of-the-art methods.

Chapter 2

Related Work

In this chapter we review the related literature and put our contribution in context to the existing approaches. Although 3D hand pose estimation is similar to other structured keypoint estimation methods, such as facial landmark or body pose estimation, we limit our discussion to the relevant works in the field of hand pose.

2.1 Generative vs Discriminative Methods :

Hand pose estimation methods can be broadly divided into generative and discriminative methods. While both approaches aim to assign a pose in an output space to an observation in the input space, the way this assignment is performed is fundamentally different.

2.1.1 Generative Methods :

Generative methods [2–4] adopt a hand model based on the kinematic structure of hands, employs a similarity function that measures the fit of the observed image to the model, and uses an optimization algorithm that maximizes the similarity function with respect to the model parameters. Generative methods also require an initialization for the pose, for example the pose in the previous frame. If the hand mesh model being used is good, generative models can produce very accurate results.

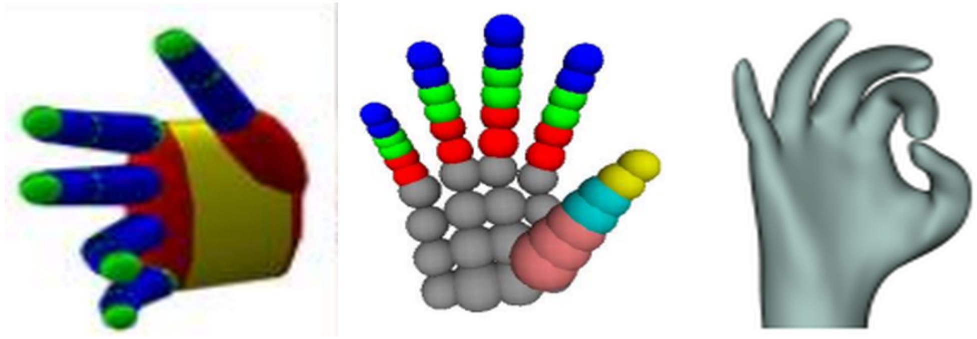

Figure 2.1: Hand models used by [2–4] respectively

A large number of hand models were proposed that used different hand crafted geometric approximations of the hand and some popular models are shown in the following figure. In order to work well, these hand models should be adjusted to the users hands, which requires an additional optimization step or manual adjustment of hand shape parameters. Different modalities were proposed for the similarity function using edge detection, optical flow, object silhouettes, shading and texture, salient points with various amounts of success. And finally optimiziation of the similarity function is a critical issue as the high dimensional pose space is prone to local minimas. Particle Swamp Optimization(PSO) [2], Iterative Closest Point(ICP) [18], or their combination (ICP+PSO) [6] have been used to optimize the hand points together with kinematic and temporal constraints.

2.1.2 Discriminative Methods:

The second type of approach is based on discriminative models that aim at directly predicting the joint locations to the discrete or continuous parameter space using color, depth or RGB images. The accuracy of these methods rely critically on the mapping from image to pose.

Some approaches segment the hand parts first and estimate the pose in a second step. Xu *et al.* [5] assign each pixel a hand part label and then infers the 3D hand joint locations using a 3 stage pipeline. Direct regression based methods [6, 7] estimate a subset of the parameters directly without intermediate representation. Generalization in terms of capturing illumination, articulation and view-point variations can be achieved only through adequate representative training data. But acquisition and annotation of realistic training data is a difficult and costly procedure. For this reason most approaches rely on synthetic rendered data that has inherent ground-truth annotations [19].

2.2 Different Sensor Based Hand Pose Estimation

Vision based hand pose estimation methods that use RGB or depth cameras are common in the field of Computer Vision [3, 6, 8–10, 17]. However, camera based methods suffer from various depth and pose ambiguities due to occlusion and self-similarity of hands. To circumvent these problems, other researchers have used magnetic sensors or marker-based approaches to infer the 3D human poses [11, 12]. There are also task-tailored devices such as Leap Motion sensors that have been introduced as well. In this section, we provide a brief overview of the different sensors being used in monocular or multi-view camera setups and the datasets created using these sensors.

2.2.1 RGB based Hand Pose Estimation

Pioneering works [20, 21] in the field of hand pose estimation used RGB images to estimate 3D hand poses. Gorce *et al.* proposed a model that estimates 3D hand pose, texture and illuminant dynamically. They used two synchronized, calibrated cameras to obtain ground truth 3D measurements on the hand. Vision based reconstruction of the 3D pose of human hands from RGB images is really difficult since any given 2D point in the image plane can correspond to multiple 3D points in the world space. So, 3D hand

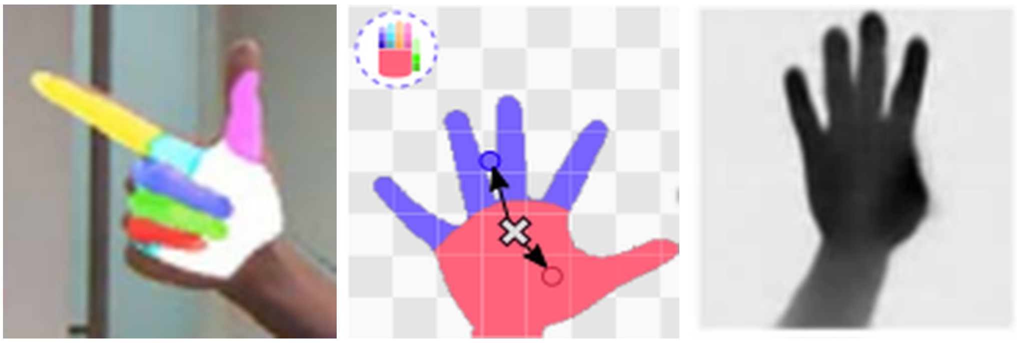

FIGURE 2.2: Examples of some discriminative hand pose estimation methods [5–7] respectively

pose estimation from RGB images suffer from this depth ambiguity in addition to the other challenges related to hand pose estimation.

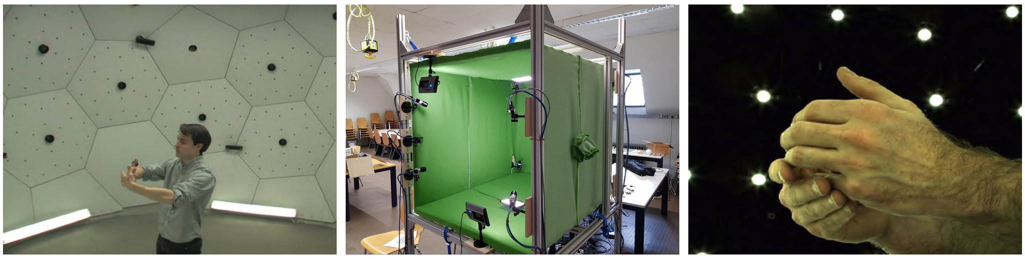

FIGURE 2.3: Multi-view RGB camera setups for hand pose estimation by [8], [9], [10] respectively

To tackle this issue, several works have proposed multiple RGB camera settings where, if the hands are occluded from one camera view, the pose can be correctly inferred from other unoccluded views. Simon *et al.* [8] used 31 HD cameras in a Panoptic Studio setup to capture the same hand poses from multiple viewpoints. Using views of the hand where keypoint detection is easy, they use these detected keypoints to triangulate the 3D position of the hand joints. Difficult views with failed detection are annotated using the reprojected 3D keypoints. Following the same idea, Zimmermann *et al* [9] used 8 calibrated and temporally synchronized RGB cameras to create the FreiHAND dataset. Recently, Moon *et al.* [10] also created a dataset that captures single as well as more challenging interacting hand poses in a multi-view studio setup with 80 to 140 high resolution cameras. These datasets are very large in size due to being captured from multiple different viewpoints. The camera setups used for creating these datasets have been illustrated in Figure 2.3.

2.2.2 Depth based Hand Pose Estimation

Depth cameras using Time-of-flight(ToF) technology can be used to measure depth using a single camera. A ToF camera uses infrared light to determine the depth of the objects in front. Using the time it takes the signals to hit the object and bounce back, it can accurately measure distance of the object from the camera. Intel's Creative Interactive Gesture Camera was used to create several benchmark hand pose datasets [3, 6, 14], while another popular work [17] used the PrimeSense Carmine 1.09 depth camera to create the NYU dataset.

Some of these hand pose datasets use generative approaches [6, 14] to fit a pre-defined hand model to the input depth map by minimizing some hand-crafted cost functions. Discriminative approaches, which directly localizes hand joints from an input depth map, has also been proposed. Tompson *et al.* [17] used a deep neural network based method to estimate the 2D heatmaps for each hand joint. Ge *et al.* [22] extended this work by estimating 2D heatmaps in multiple viewpoints, and Guo *et al.* [23] proposed a tree-structured Region Ensemble Network to estimate the 3D hand coordinates correctly.

2.2.3 Magnetic Sensor Based Hand Pose Estimation

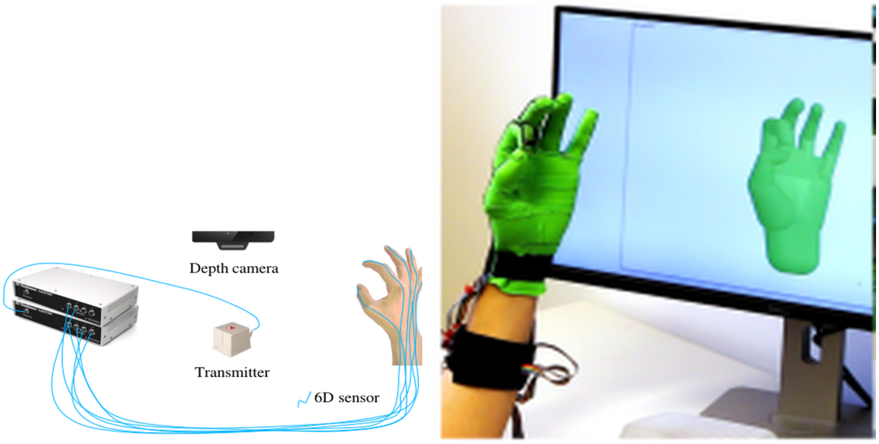

FIGURE 2.4: Magnetic Sensor based 3D hand pose estimation methods [1, 11]

In order to generate highly accurate hand-pose annotation, Yuan *et al.* used six 6D magnetic sensors to automatically create a million-scale dataset [1]. As shown on figure 1.3, global hand pose is inferred from the location and orientation of these 6 sensors, one on the back of the palm and five other sensor on each finger nail. The fingernail sensors are used to infer the TIP, DIP and PIP joints of the corresponding finger using bone lengths and physical constraints. There are other methods such as Glauser *et al.* that used stretch-sensing soft hand gloves to capture the hand pose with precision [11]. The glove consists of a full soft composite of a stretchable capacitive silicone sensor array and a thin custom textile glove(Figure 2.4) and can capture hand poses with 25 degrees of freedom.

Hand pose estimation methods using magnetic sensors work in diverse and challenging settings such as highly occluded poses or changing light conditions. However, these methods require magnetic sensors to be placed at each finger joint to capture the hand pose and the sensors need to be small enough to be wearable. But the sensitivity of magnetic sensors increases with their size, meaning small sensors lack in precision and are easier to be disturbed by external magnetic fields [24]. Also, magnetic sensors are not very practical to use in day-to-day operations and sometimes restricts the free articulation of human hands.

2.3 Methods for Hand Pose Annotation

Different public hand pose datasets have been published in recent years that uses different annotation methods. Some of the earlier works on depth based hand pose annotation used tracking based methods to annotate the frames. Some researchers created manually annotated datasets to evaluate other methods compared to ground-truth but these datasets are limited in the number of frames. Synthetic hand pose datasets and marker-based hand pose datasets have also been used for dealing with challenging hand articulations. Additionally, multi-view RGB camera based methods have used semi-automatic methods to annotate hand-poses in difficult views using 3D reprojection from unoccluded viewpoints. Rogez *et al.* also introduced a semi-automatic hand pose dataset using egocentric depth images and used a semi-automatic labelling tool to annotate frames with hand-object interaction. Some of these datasets contain static poses only, while others contain complex hand articulations. Also, some datasets only consists of hand poses with single hand, while others contain hand-object interaction or interaction between both hands. However, we are mainly interested in the method being used to annotate the hand poses and therefore skip this detail. In the following table 2.1, we show some state-of-the-art datasets, the annotation method being used and other characteristics of the datasets. We discuss these annotation methods in more detail in the following discussions.

Dataset	Annotation Method	Source	resolution	subjects	frames
ICVL [6]	tracking+refinement	Depth	320X240	10	180k
MSRA15 [14]	tracking+refinement	Depth	320X240	9	76k
NYU [17]	tracking+refinement	Depth	640X480	10	180k
Dexter+Object [16]	manual	RGB-D	640X480	1	3k
EgoDexter [15]	manual	RGB-D	640X480	4	1.5k
STB [25]	manual	RGB-D	640X480	1	18k
BigHand2.2M [1]	marker-based	Depth	640X480	10	2.2M
FPHA [26]	marker-based	RGB-D	1920X1080	6	105k
RHD [13]	synthetic	RGB-D	320X320	20	44k
SynthHands *et al.* [27]	synthetic	RGB-D	320X240	5	80k
Rogez *et al.* [28]	semi-automatic	RGB-D	n/a	8	12k
FreiHAND [9]	semi-automatic	RGB	224X224	32	134k
Simon *et al.* [8]	semi-automatic	RGB	1920X1080	n/a	15k
InterHand2.6M [10]	semi-automatic	RGB	512X334	27	2.6M

Table 2.1: Comparison of Existing Datasets using different annotation methods

2.3.1 Manually Annotated Hand Pose Datasets

Creating datasets using manual annotation is a challenging and labor-intensive approach. These benchmarks are generally small in size due to the difficult of annotating them. MSRA14 [3] dataset was created using six subjects who perform various rapid gestures. A 400-frame video sequence is recorded for each subject and the ground truth hand poses were manually labelled for all 2400 frames. Other datasets such as Dexter+Object [16] provide 3k frames with ground truth annotations while the EgoDexter [15] consists of 1485 frames of ground truth 2D and 3D fingertip positions. These datasets were mainly created for evaluation purposes and creating a large dataset using manual annotations is just not feasible.

2.3.2 Synthetic Hand pose Datasets :

Hand pose datasets that capture real hand images are limited in quantity and coverage. Additional synthetic data can be used to increase the accuracy of 3D pose estimation. Compared to real datasets, it is easier to acquire synthetic data. Synthetic frames can be used to create virtually infinite training data with large variations in shapes and view-points and produce annotations that are more accurate in case of occlusions [29].

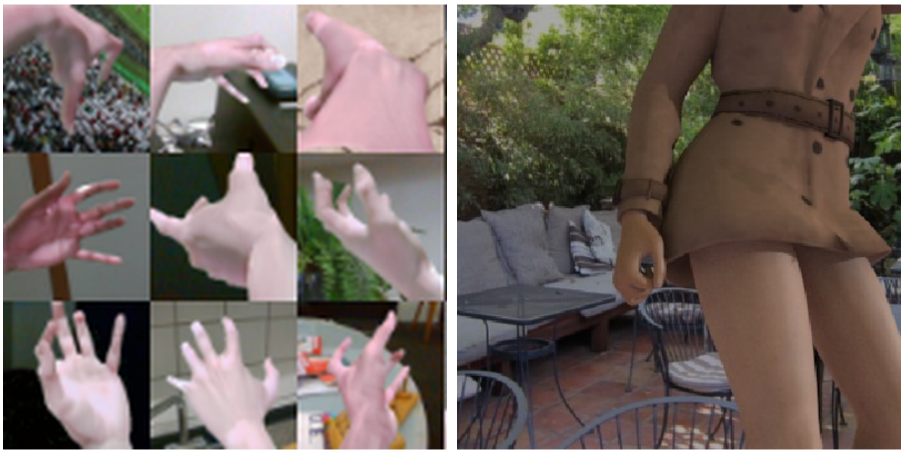

Figure 2.5: Synthetic hand pose datasets show shape and appearance variance from real hand images[12, 13]

For these reasons, multiple synthetic image datasets have been introduced in recent years [12, 29] that contain 300k to five million frames. However, synthetic hands exhibit a certain level of deviation from real images as these do not capture the sensor characteristics such as noise and missing data that are present in real images. Also, synthetically generated images sometime produce kinematically implausible and unnatural hand poses. We show some examples of these synthetic datasets in Figure 2.5.

2.3.3 Tracking Based Hand Pose Datasets

With the introduction of Microsoft Kinect, which uses a color sensor and a depth sensor to capture RGB images with associated depth, the tasks of hand detection and segmentation have been simplified. The ICVL dataset [6] is one of the first benchmark datasets on depth images that uses 3D skeletal tracking followed by manual refinement. The dataset contains 180k training images from 10 different subjects, however the limitations of annotation accuracy have been noted in literature [14]. MSRA15 [14] is one of the more complex datasets in the field that consists of 76,500 depth images captured from 9 subjects, using Intel's Creative Interactive Camera. It is annotated in an iterative way where an optimization method [3] and manual re-adjustment procedure alternate until convergence. These annotations are also reported to contain annotation errors such as missing finger and thumb annotations [30]. Tompson [17] used 3 RGBD sensors at viewpoints separated by approximately 45 degrees surrounding the user from the front

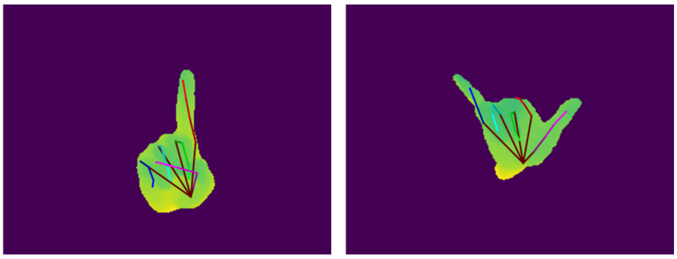

FIGURE 2.6: Annotations errors in the MSRA15 Dataset [14]

to create the NYU hand pose dataset. They used a predefined 3D hand model that was manually readjusted for poses that failed to fit perfectly. The dataset contains 80k depth frames and regarded as one of the most popular benchmarks for 3D hand pose estimation.

Tracking based hand pose annotation fails to generate accurate hand poses when there is a good amount of occlusion in the hand pose. In addition, the datasets [6, 14, 17] are prone to tracking based failures and requires some amount of manual refinement. We show some noisy annotations from MSRA15 dataset in the Figure 2.6

2.3.4 Automatic Hand Pose Annotations :

Additional sensors such as data-gloves or magnetic sensors can aid automatic capture of human hands [1, 31], but care must be taken not to restrict the natural motion of human hand. The ASTAR dataset used a ShapeHand data-glove which has been reported to influence the captured hand images, and to some extent hinder free hand movement. Less intrusive magnetic sensors [32] have been proposed, however, they only provide finger tip annotations. Also, some data glove annotations are not very accurate and would be visible in training images which biases the learning algorithm.

2.3.5 Semi-automatic Hand Pose Annotations :

Semi-automatic approaches of hand pose annotation has recently been used to create large scale hand pose datasets. Moon *et al.* [10] used semi-automatic approach for creating a multi-view RGB dataset, which combines manual annotation with automatic machine annotation and proposes a much more efficient approach compared with manual annotation. Using their semi-automatic method using multi-view RGB images, they achieve very small annotation errors(2.78 mm) compared with fully manual annotation. Simon *et. al* [8] also proposed a multi-view RGB dataset, that uses multiview Bootstrapping to accurately annotate difficult or occluded hand poses. It uses easily detectable views to triangulate the 3D keypoints and using the reprojected 3D keypoints difficult views are annotated correctly.

Rogez *et al.* [28] proposed a semi-automatic method for hand pose estimation from egocentric viewpoints. They used a chest-mounted RGB-D sensor to collect egocentric hand-object interactions and propose a semi-automatic labelling tool to annotate partially occluded hands and fingers in 3D. Oberweger *et al.* extended this work by creating a semi-automatic pipeline for annotating all the frames in a hand depth video using manual 2D annotations in some of the frames. They use manual annotation of 2D joints in approximately 10% frames to infer the 3D hand pose annotations for all the frames in a depth video sequence. Compared to this work, we propose a method that selects the number of frames to be annotated in an automatic manner, minimizes annotation work by selecting fewer number of frames and provides more accurate 3D annotation compared with other state-of-the-art methods.

Chapter 3

Proposed Method

Given a sequence of N depth frames $\{D_i\}_{i=0}^{N}$ capturing a hand in motion, our goal is to estimate the 3D hand joints in the depth maps while eliminating as much manual effort as possible. Our approach is based on the common observation that not all the frames in a hand depth video vary significantly from each other. Therefore, if we annotate the hand joints in some frames and propagate this information to the similar frames, that should lead to better accuracy. The process starts by automatically selecting the frames for manual annotation. We propose to select a small subset of frames such that all the other frames are within a distance threshold from the selected frames. We refer these small subset of selected frames as reference frames. A human annotator provides the 2D hand joint locations in the reference frame and using these we infer the 3D position of hand key-points in the reference frames. We propagate this knowledge to the other frames that are similar to the corresponding reference frames and infer the 3D hand joints in the remaining frames. Finally, we perform a global optimization enforcing appearance, temporal and spatial constraints to further optimize the hand joint locations.

3.1 Selecting the Reference Frames

We start by selecting the reference frames to do manual annotation. A simple way to select the reference frames would be to regularly sample the video after a time interval, for example every n-th frame could be selected as a reference frame. However, this

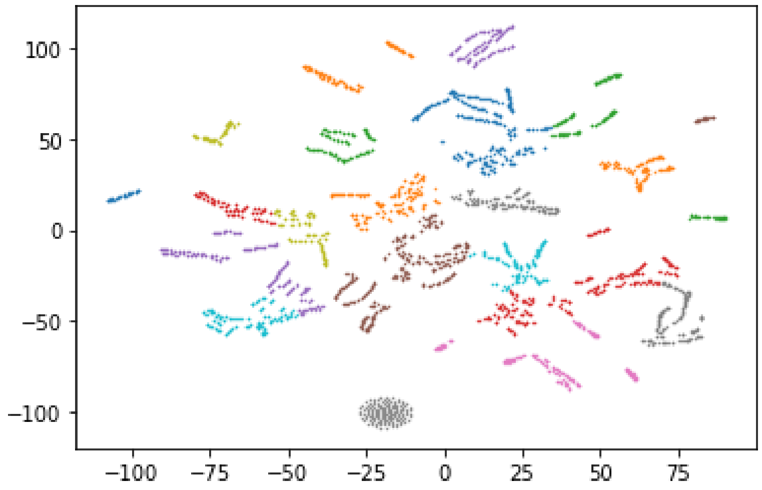

Figure 3.1: TSNE plot for Blender Dataset

type of selection process might not be optimal for annotating hand poses. Since hand movement can be fast or slow in different part of the video and therefore selecting a definite value of n might not work for the whole sequence. For example, if we select every 10-th frame as a reference frame, in some cases we will find that the hand pose changed a lot over those 10 frames and selecting one of those 10 frames as reference would not be a good representation for the other 9 frames. On the other hand, users tend to keep their hand still in between poses and there could be a lot of consecutive frames that are almost same but due to selecting every 10-th frame as reference, we could end up selecting very similar frames. Therefore, this kind of temporal selection of frames won't be able to select a subset of frames that are representative of all the frames in the video sequence.

Instead of temporal sampling, we would like to select the reference frames in such a way that for each unannotated frame, there is a minimum degree of similarity with one of the annotated frames, using which the 3D pose of the unannotated frame can be inferred more accurately. Also, another constraint is that we would like to select as few reference frames as possible to save time annotating them. In other words, we need to find groups or clusters in the dataset frames so that frames belonging to each group are similar and we can select one of those frames for doing manual annotation work. We can use TSNE (t-distributed Stochastic Neighbor Embedding) [33] to visualize very high dimensional data in a low-dimensional space to identify relevant patterns. The main advantage of

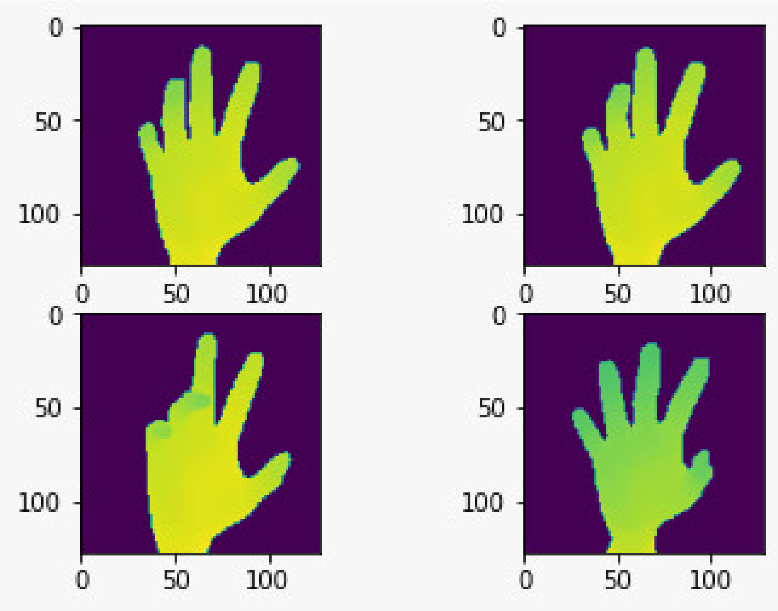

FIGURE 3.2: Visualization of frames that belong to the same TSNE cluster

TSNE is the ability to preserve local structure. This means that points which are close to one another in the high-dimensional data set will tend to be close to one another in the low dimensional plot as well. We visualize the 3040 frames in the Blender Dataset [34] as points in the TSNE plot 3.1. Each colored dot represents a frame and the color encodes the temporal order. Temporal changes of the hand articulation can be clearly observed from the different trajectories.

The clusters in the TSNE plot clearly suggest that, there are a number of frames that are very similar to each other. In Figure 3.2 we show a few frames belonging to the same cluster in the TSNE plot and these frames are indeed quite similar to each other. So, the idea is to select one or few of these frames to use as reference to estimate the pose in the other frames in the cluster. However, TSNE does not preserve distance relationships in the lower dimensional space, meaning points next to each other in the TSNE plot might not be the closest match in the high dimensional space. So, we can not take the centroid of our TSNE embedding to select our reference frame for that cluster. Also, we can not assume the relative size of clusters from the TSNE plot as TSNE tends to expand dense clusters and shrink sparse ones. But TSNE confirms our assumption that some of the frames are indeed similar to each other and we need a similarity measure to pick some of these frames as reference frames.

To find the similarity between frames, we define a distance function on the depth frames. We propose to use cosine distance to measure the similarity between frames and use ρ as a threshold for minimum similarity.

$$\forall i \; \forall j \;\; s.t. \; i \neq j, d(D_i, D_j) = cos(\theta) = \frac{D_i.D_j}{||D_i||||D_j||} \tag{3.1}$$

Here, D_i and D_j are two depth maps, and they are considered similar if d(D_i,D_j) $< \rho$. We start by selecting the first frame as a reference frame and iterate over the whole set of frames. Every time the distance of the next frame is greater than the threshold ρ, the next frame is considered as the reference frame. If there are other frame within ρ distance of a reference frame then we know that these frames are significantly similar to the selected reference frame and therefore we do not need manual annotation for this. This is illustrated by the equation 3.2 below

$$\mathrm{R} = \begin{cases} 1 & if \; d(D_{i-1}, D_i) > \rho \\ 0 & else \end{cases} \tag{3.2}$$

Here, D_{i-1} is the previous frame and D_i is the current frame being compared. Selecting the reference frames in this way has advantages over temporal sampling as consecutive frames could have much higher distance than a frame that occurs much later in the video sequence. Also, we see from the preliminary results that we achieve better results than the greedy reference frame selection method of Oberweger *et al.* [30]. Moreover, we can change the ρ threshold to change the number of reference frames. A low threshold would increase the number of reference frames, this would not harm the accuracy of the annotations per say but would drastically increase manual annotations. On the other hand, a high threshold would pick far too less frames which would not be representative of the entire dataset and could yield sub-par results.

3.2 Initializing the 3D Joint Location in the Reference Frames

After selecting the reference frames we need to label them by a human annotator. We use the annotation tool by Oberweger *et al.* [30] to annotate the frames. The annotator provides the 2D hand joint locations for each reference frame alongside the visibility information. The visibility information basically points whether the joints are closer or farther from the camera than the parent joint in the hand skeleton tree. Using this information we can recover the 3D locations of the joints. To recover the 3D locations of the joints in the reference frame, we optimize the following non-linear least squares problem used by Oberweger *et al.* [30].

$$\underset{\{L_{r,k}\}_{k=1}^{K}}{argmin} \sum_{k=1}^{K} v_{r,k} ||proj(L_{r,k}) - l_{r,k}||_2^2 \tag{3.3}$$

$$\begin{aligned} s.t.\ \forall k\ ||L_{r,k} - L_{r,p(k)}||_2^2 &= d_{k,p(k)}^2 \\ \forall k\ v_{r,k} = 1 &\implies D_r[l_{r,k}] < z(L_{r,k}) < D_r[L_{r,k}] + \epsilon \\ \forall k\ v_{r,k} = 1 &\implies (L_{r,k} - L_{r,p(k)})^T\ .\ c_{r,k} > 0 \\ \forall k\ v_{r,k} = 0 &\implies z(L_{r,k}) > D_r[L_{r,k}] \end{aligned}$$

where, r = the index of the reference frame. $v_{r,k} = 1$ if the k-th joint is visible in the r-th frame. $L_{r,k}$ = 3D location of the k-th joint in the r-th frame. $l_{r,k}$ is the 2D reprojection of the 3D joint locations. proj(L) returns the 2D reprojection of a 3D location. p(r) returns the index of the parent joint of the k-th joint in the hand skeleton. $d_{k,p(k)}$ is the known distance between the k-th joint and it's parent p(k). $D_r[l_{r,k}]$ is the depth value in D_r at location $l_{r,k}$. z(L) is the depth of 3D location L. ϵ is a threshold used to define the depth interval of the visible joints. In practise, we use $\epsilon = 15$ mm given the physical properties of the hand. $c_{r,k}$ is equal to the vector $[0, 0, -1]^T$ if the k-th joint is closer to the camera than it's parent in the frame r, and $[0, 0, 1]^T$ otherwise. $(L_{r,k} - L_{r,p(k)})^T$ is the vector between joint k and it's parent in the frame.

The constraints in Equation 3.3 assure that (1) we find the 3D joints $L_{r,k}$ such that the bone lengths i.e. the distance between 2D projection of $L_{r,k}$ and $l_{r,k}$ is maintained; (2) visible joints are within ϵ distance of observed depth maps; (3) the z-azis value for hidden joints is greater than visible joints, and (4) depth order constraints between a joint and it's parent is also maintained. We assume the lengths $d_{k,p(k)}$ are known. During implementation, we calculate this distances as the euclidean distance between joints in the hand skeleton tree.

We use SLSQP[35] to solve this problem and find the the 3D hand keypoints i.e. $L_{r,k}$ values. Finding the hand keypoints in this way maintains the constraints of the hand skeleton tree and provides a reasonable estimate of the 3D hand joints in the reference frames.

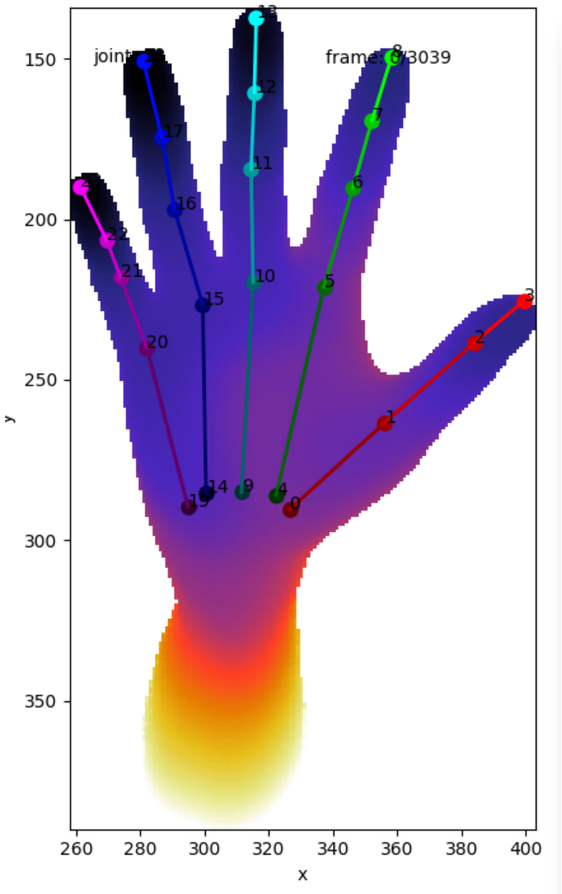

FIGURE 3.3: A human annotator marks the 2D hand joint locations in the depth frames

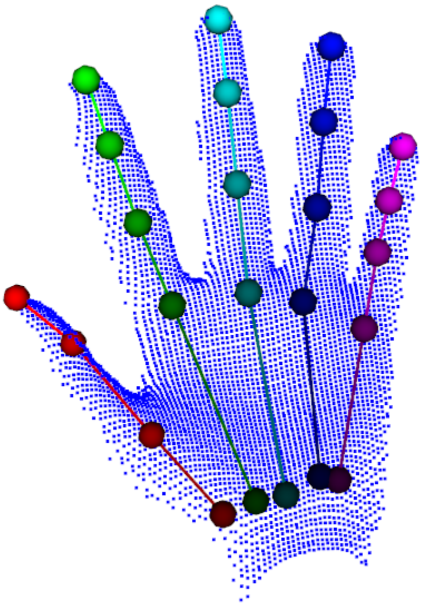

FIGURE 3.4: Visualization of the 3D hand keypoints based on the 2D annotations

3.3 Initializing the 3D Joint Locations in the Remaining Frames

The previous section computes the 3D location of hand joints in the reference frames. Now we need to propagate this information to the remaining frames to improve the accuracy of their 3D annotations. After propagating the 3D joint information from a reference frame and estimating 3D joint locations in a non-refernce frame, we do not add the newly annotated frame to the list of reference frames as was done by Oberweger *et al.* [30]. The reason is that the reference frames are annotated by humans and we can be very confident about their 3D joint locations. However, any frame using this reference frame may estimate it's joints locations poorly and we don't want a frame with possible errors to be added to the list of reference frames. We have already figured out the closest reference frame for each of the remaining frames in the cosine distance calculation part described in section 3.1 and therefore we can infer the 3D joint locations of the remaining frames using their closest reference frame. This way, we make sure that the human annotated frames are being used to infer the joint locations in the remaining frames and by not allowing estimated frames to be added to the list of reference frames, we maintain the accuracy of 3D joints in the reference frames.

The procedure is shown using 3.4 where, I is the set of reference frames for which the 3D location of the joints have been initialized. So, we check for a frame $\hat{c}$ not initialized yet and find it's closest reference frame $\hat{a} \in I$ to estimate it's joints.

$$\begin{bmatrix} \hat{c} \\ \hat{a} \end{bmatrix} = \underset{c \in [1;N]; a \in I}{argmin} \; d(D_c, D_a) \tag{3.4}$$

We use the appearance of joints in $\hat{a}$ to predict their 3D locations $L_{\hat{c},k}$ in $\hat{c}$ by minimizing equation below. Here, $ds(D_{\hat{c}}, proj(L_{\hat{c},k}); D_{\hat{a}}, l_{\hat{a},k})$ denotes the dissimilarity between the patch in D_1 centered on the projection L_1, proj(L_1) and the patch D_2 centered on l_2. This optimization looks for joints based on their appearance in frame $\hat{a}$ while enforcing the 3D distances between the joints. We use Levenberg-Marquardt algorithm to solve this following [30].

$$\begin{aligned} &\sum_{\{L_{\hat{c},k}\}_k} \sum_k ds(D_{\hat{c}}, proj(L_{\hat{c},k}); D_{\hat{a}}, l_{\hat{a},k})^2 \\ &s.t. \; \forall k \; ||L_{r,k} - L_{r,p(k)}||_2^2 = d_{k,p(k)}^2 \end{aligned} \tag{3.5}$$

To align the nearest frame we use SIFT-Flow [36]. Unlike Optical flow which aligns an images to it's temporally adjacent frame, SIFT-Flow aligns an image to it's nearest neighbors in a large corpus containing a variety of poses. It matches densely sampled, pixel-wise SIFT features between two images, while preserving spatial continuities. We use the appearance of joints in reference frames to predict their location in the non-reference frames by minimizing the dissimilarity.

At each iteration, a non-reference frame is processed using it's closest reference frame $\hat{a}$ ∈ I. First, we initialize the SIFT-Flow by aligning the closest reference frame $\hat{a}$ to a non-reference frame $\hat{c}$. This maps the 2D reprojection of joints in frame $\hat{a}$ to 2D locations in frame $\hat{c}$. We backproject each of these 2D joint locations on the Depth map D_c to initialize $L_{\hat{c},k}$. We check for each 3D joint in the non-reference frame i.e. $L_{\hat{c},k}$, the distance to it's parent joint $L_{\hat{c},p(k)}$ and thereby enforcing the bone joint constraints in the hand skeleton tree.

3.4 Global Optimization

The previous optimization already optimizes the frames based on their closest reference frames. However, there might be some hand constraint violations due to estimating the hand joints in remaining frames using their closest reference frames. We also maintain some temporal constraints with the previous frame. We perform a global optimization over the 3D joint locations $L_{i,k}$ for all the frames by minimizing the equation below using the method by Oberweger *et al.* [30]:

$$\begin{aligned}
&\sum_{i\in[1;N]\backslash R}\sum_{k} ds(D_i, proj(L_{i,k}); D_{\hat{i}}, l_{\hat{i},k})^2 + && (C)\\
&\lambda_M \sum_{i}\sum_{k} ||L_{i,k} - L_{i+1,k}||_2^2 + && (TC)\\
&\lambda_P \sum_{r\in R}\sum_{k} v_{r,k} ||proj(L_{r,k} - l_{r,k})||_2^2 && (P)\\
&s.t. \forall i, k ||L_{i,k} - Li, p(k)||_2^2 = d_{k,p(k)}^2
\end{aligned}$$

The first term (C) sums the differences of the joint locations compared to the closest reference frame. Given the depth map and 3D joint locations in the current frames, it calculates the dissimilarities with the closest reference frame. The second term (TC) is a temporal constraint that makes sure that consecutive joints do not have huge fluctuations between their 3D joints. Because, hand pose from consecutive frames can not change very rapidly, this term maintains temporal smoothness by avoiding consecutive joint estimations that are far away from each other. The last term(P) of the summation ensures consistency with the manual 2D annotations for the reference frames since the 2D reprojection of 3D hand joints should be similar to what the user annotated in 2D. λ_M and λ_P are weights that maintains the significance of each constraint. Using these weights with their corresponding constraints, we make sure that the 3D hand joints maintain the shape and temporal constraints and therefore this global optimization step further refines the annotations.

Chapter 4

Experiments

In this chaper, we demonstrate the experimental results of our proposed semi-automatic hand pose estimation method [37]. We start with selecting one real and one synthetic hand pose dataset. For experiments with semi-automatic annotation, we choose a state-of-the-art real hand pose dataset, MSRA15 [14] and a synthetic hand pose dataset(Blender), that was introduced by [34]. We apply our reference frame selection method on these datasets and show that our reference frame selection covers different hand pose articulations using the TSNE diagram. We also show our preliminary results on the Blender dataset compared to another state-of-the-art hand pose estimation method [30]. Finally, we discuss our results using the proposed semi-automatic hand pose annotation pipeline and define our future scope of work.

4.1 Evaluation on a Synthetic Hand Pose Dataset

We start by showing our results on a synthetic hand pose dataset called Blender. The Blender dataset contains 3040 frames of single hand articulations. We apply our proposed reference frame selection method based on cosine distance with a distance threshold ρ. Using $\rho = 0.035$, we selected 204 reference frames from the 3040 frames of the Blender dataset. We plot the TSNE diagram with the selected reference frames in Figure 4.1. The TSNE embedding for all the frames in the Blender dataset is shown in 'Blue' and the selected reference frames is marked in 'Orange'. The points in the TSNE plot that form a line implies the temporal similarity between those frames. For each of these lines, we would like to select at least one frame that is similar to the other frames in that line. From the observation, it seems that the frames selected using cosine similarity covers each of the lines in the TSNE plot for the Blender dataset. So, we can say that our selected reference frames are representative of all the frames of the Blender dataset.

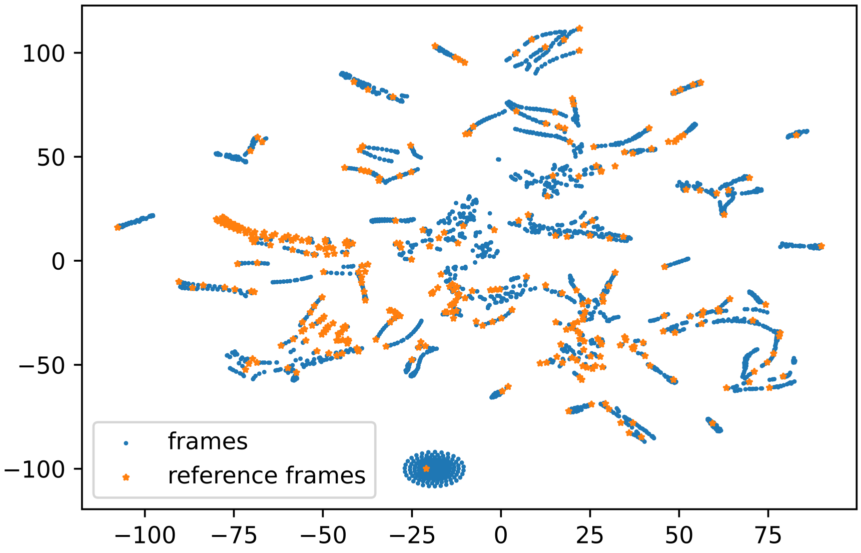

Figure 4.1: Proposed reference frame selection for the Blender hand pose dataset

After selecting the reference frames, we will have to annotate the 2D keypoint locations in these reference frames. Since Blender is a synthetic hand pose dataset, the joint locations are highly accurate and therefore we use the Blender ground truth annotations as our manual 2D annotation. We only use the ground truth joint location of the reference frames and try to estimate the joint locations in the other remaining frames. Finally, we compare our estimated 3D keypoints with the ground truth 3D keypoints and measure the mean, median and maximum error. We also compare our results with the method of Oberweger *et al.* [30] and show the results on Table 4.1. The preliminary results show that we achieve lower mean and median error than [30] and therefore we can say that our proposed reference selection has lead to this improvement in accuracy.

	Our Method	Oberweger *et al.* [30]
mean error (mm)	4.69	4.91
max error (mm)	84.33	73.65
median error (mm)	3.35	3.68

Table 4.1: Comparison of Final Results on the Blender Dataset

4.2 Evaluation on Different Number of Reference Frame Selection

We can select a different number of reference frames based on a different cosine distance threshold ρ. If we pick a higher threshold, ρ we will get a lower no of reference frames which will save a lot of manual annotation work. Also, if we want better accuracy and have resources to annotate more reference frames then we can reduce the value of ρ and then we will select a higher number of reference frames to be annotated. We can compare the results of selecting higher or lower number of reference frames from the following table :

	∼2% frames (73 frames)	∼5% frames (141 frames)	∼10% frames (288 frames)	Oberweger *et al.* [30] (304 frames)
mean error (mm)	5.79	5.50	4.69	4.91
max error (mm)	76.79	78.98	84.33	73.65
median error (mm)	4.60	4.17	3.35	3.68

Table 4.2: Evaluation on the Blender Dataset for different % of reference frame selection

As we can see from the table, the annotation results improved with the selection of more reference frames, i.e. with more human annotation work we can always get better results. However, we got comparably good results with Oberweger *et al.* [30] while selecting 5% of the frames as reference frames and our results are better than [30] when we consider approximately same number of reference frames. This shows that with small effort of annotating only a fraction of the frames with can get comparably good results with ground truth annotation.

4.3 Evaluation on a Real Hand Pose Dataset

Using our proposed method on reference frame selection, we improved the annotations of a real hand pose dataset. The MSRA15 hand pose dataset is one of the benchmark datasets for depth based hand pose estimation. It contains 9 subjects performing 17 different signs in front of single depth camera. Each sign was performed for 500 frames and therefore for the total 17 signs, the dataset contains 17*500 = 8500 frames for each user. For the total 9 subject, the total no of frames = 8500*9 = 76500. The dataset was generated using hand tracking followed by some manual refinement. As tracking based methods are prone to tracking failures, the annotations are not very accurate as can be seen from the figure below.

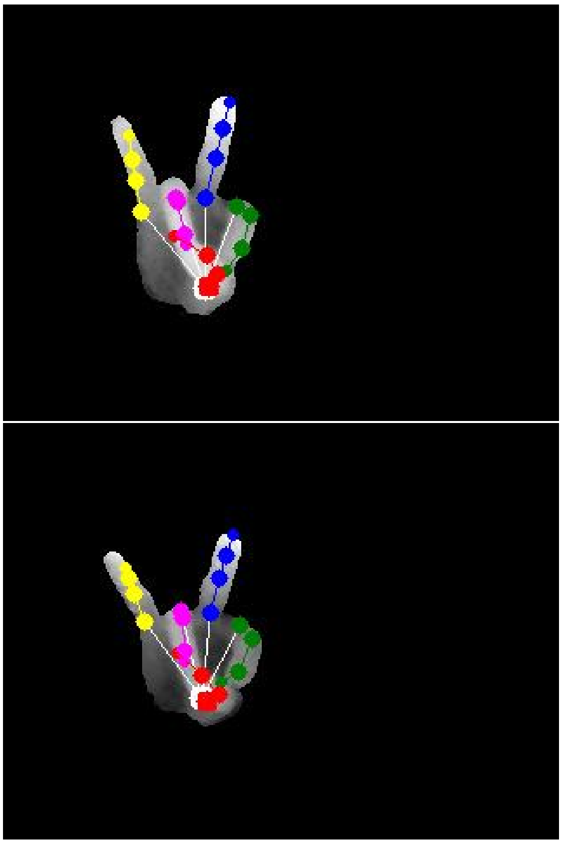

Figure 4.2: Ground Truth Annotations of the MSRA dataset

Using our semi-automatic approach of hand pose annotation we will select a small subset of reference frames from the 8500 frames for a particular subject. Then, we manually annotate the 2D joints of the selected reference frames. Using the manual supervision provided by the 2D annotations, we achieved better 3D hand pose estimation results for the whole dataset. Using the 2D annotation tool, the users provides the 2D joint location for each of the 21 hand joints and whether the joints are visible or not in that frame. Using SLSQP [35] we obtain the 3D joint locations of the reference frames and by aligning the reference frames with other frames using SIFTFLOW [36], we can infer the 3D keypoint locations in all the remaining frames. These 3D keypoints are then further optimized using the Global Optimization step which enforces temporal smoothness and ensures consistency with the manual annotations. Since we do not know ground truth 3D joint locations on real datasets, we can visualize the qualitative results using our proposed semi-automatic method. Therefore, we show some qualitative results on MSRA15 dataset on 4.3.

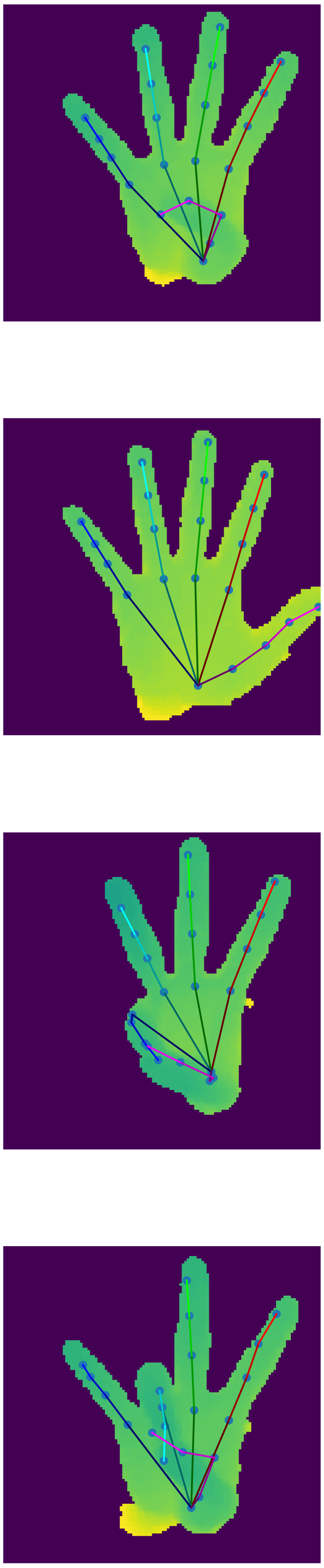

Figure 4.3: Qualitative Results on the MSRA dataset

Chapter 5

Conclusion

Training data is the backbone of the deep learning methods being used for hand pose estimation. For datasets with real images, it is very difficult to get accurate 3D joint locations due to noise, self-occlusion and complexity of human hand structure. This annotation errors are also present in state-of-the-art hand pose datasets as shown in Figure 2.6. Our proposed method provides a solution by processing some frames with manual supervision and propagates this information to the other frame to get more accurate annotations. This saves time required to manually annotate all the frames and provides better accuracy than inferring the annotations without any manual supervision. Moreover, this pipeline of annotating frames using a representative subset can be applied for other articulated structures such as human bodies or other relevant annotation tasks. Finally, we have demonstrated our semi-automatic method on hand pose datasets containing single hand images and where hand segmentation was easy. As a future work, it remains to be seen whether this method can be extended to the more challenging double-handed poses or for annotating hand-object interactions.

Bibliography

[1] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017.

[2] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011.

[3] 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, 2014.

[4] 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): 1–12, 2016.

[5] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3462, 2013.

[6] 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, 2014.

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

[8] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[9] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019.

[10] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. *arXiv preprint arXiv:2008.09309*, 2020.

[11] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

[12] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018.

[13] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.

[14] 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, 2015.

[15] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an

egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1284–1293, 2017.

[16] 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 *European Conference on Computer Vision*, pages 294–310. Springer, 2016.

[17] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[18] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, volume 34, pages 101–114. Wiley Online Library, 2015.

[19] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016.

[20] 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.

[21] 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.

[22] 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, 2016.

[23] Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. Region ensemble network: Improving convolutional network for hand pose

estimation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4512–4516. IEEE, 2017.

[24] Weiya Chen, Chenchen Yu, Chenyu Tu, Zehua Lyu, Jing Tang, Shiqi Ou, Yan Fu, and Zhidong Xue. A survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors*, 20(4):1074, 2020.

[25] 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.

[26] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018.

[27] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[28] Grégory Rogez, Maryam Khademi, JS Supančič III, Jose Maria Martinez Montiel, and Deva Ramanan. 3d hand pose detection in egocentric rgb-d images. In *European Conference on Computer Vision*, pages 356–371. Springer, 2014.

[29] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *2018 International Conference on 3D Vision (3DV)*, pages 110–119. IEEE, 2018.

[30] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4957–4965, 2016.

[31] Chi Xu, Ashwin Nanjappa, Xiaowei Zhang, and Li Cheng. Estimate hand poses efficiently from single depth images. *International Journal of Computer Vision*, 116 (1):21–45, 2016.

[32] Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. *arXiv preprint arXiv:1507.05726*, 2015.

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[34] Gernot Riegler, David Ferstl, Matthias Rüther, and Horst Bischof. A framework for articulated hand pose estimation and evaluation. In *Scandinavian Conference on Image Analysis*, pages 41–52. Springer, 2015.

[35] Dieter Kraft et al. A software package for sequential quadratic programming. 1988.

[36] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010.

[37] Marnim Galib, Giffy Jerald Chris, and Vassilis Athitsos. Semi automatic hand pose annotation using a single depth camera. In *International Symposium on Visual Computing*, pages 362–373. Springer, 2021.