SEMI-AUTOMATIC HAND POSE ESTIMATION USING A SINGLE DEPTH CAMERA

by

GIFFY JERALD CHRIS

THESIS

Submitted in partial fulfillment of the requirements

for the degree of Master of Science in Computer Science

at The University of Texas at Arlington

December 2020

Arlington, Texas

Supervising Committee:

Dr. Vassilis Athitsos, Supervising Professor

Dr. Farhad Kamangar

Dr. Chris Conly

ABSTRACT

SEMI-AUTOMATIC HAND POSE ESTIMATION USING A SINGLE DEPTH CAMERA

GIFFY JERALD CHRIS, M.S. CSE

The University of Texas at Arlington, 2020

Supervising Professor: Vassilis Athitsos

This paper addresses the problem of 3D hand pose annotations using a single depth camera. Although hand pose estimation methods rely critically on accurate 3D training data, creating such reliable training data is challenging and labor intensive. 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 annotator 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.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Vassilis Athitsos and Marnim Galib (Co-Author) for their immense support and guidance through my entire tenure here at The University of Texas at Arlington. I would also like to thank Dr. Farhad Kamangar and Dr. Chris Conly for being a part of my thesis committee. Lastly, I would like to thank Dr. Sajib Datta for all the help and opportunities he has provided me over the two years at UTA.

DEDICATION

I dedicate this paper to my parents Mr. Jerald Mathias and Mrs. Christilda Jerald for being a constant support throughout my life and helping me pursue my dreams. I would also like to thank Nishant Rodrigues for inspiring me to pursue my master's and last but not the least all my friends here at UTA for making my time here a memorable one.

LIST OF FIGURES (or Illustrations)

FIGURE 1: ANNOTATION ERRORS IN THE MSRA DATASET: IN THE FIRST FIGURE THE JOINTS ON THE LITTLE FINGER AND IN THE SECOND IMAGE JOINTS ON THE THUMB AND INDEX FINGER AREN'T CORRECTLY ANNOTATED..... 2

FIGURE 2: VISUALIZATION OF SIMILARITY IN THE VIDEO SEQUENCE. WE, THEREFORE, SELECT SOME FRAMES FROM THESE CLUSTERS AS REFERENCE FRAMES, ANNOTATE THEM AND USE THESE ANNOTATIONS TO IMPROVE THE OTHER FRAMES THAT BELONG TO THE SAME CLUSTER 7

FIGURE 3: INITIALIZATION OF THE 3D HAND JOINT LOCATIONS IN REFERENCE FRAMES 10

FIGURE 4: SIFTFLOW OPTIMIZATION OF THE REMAINING FRAMES BASED ON THE CLOSEST REFERENCE FRAME .. 12

FIGURE 5: QUALITATIVE RESULTS ON THE MSRA DATASET 17

FIGURE 6: GRAPH SHOWING ERRORS DEPICTED IN TABLE 4 18

FIGURE 7: T-SNE DISTRIBUTION OF REFERENCE FRAMES WITH RESPECT TO NON-REFERENCE FRAMES. 19

LIST OF TABLES

TABLE 1: TABLE OF SYMBOLS 8

TABLE 2: COMPARISON OF INITIALIZATION RESULTS ON THE BLENDER DATASET 15

TABLE 3: COMPARISON OF FINAL RESULTS ON THE BLENDER DATASET 16

TABLE 4: EVALUATION ON THE BLENDER DATASET FOR DIFFERENT % OF REFERENCE FRAME SELECTION 17

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 RELATED WORK 4

CHAPTER 3 PROPOSED METHOD 6

3.1 Selection of reference frame 6

3.2 Initializing the 3D Joint Location in the Reference Frames 9

3.3 Initializing the 3D Joint Locations in the Remaining Frames 10

3.4 Global Optimization 12

CHAPTER 4 EVALUATIONS 15

4.1 Evaluation on Synthetic Data 15

4.2 Evaluation on Real Data 16

4.3 Evaluation on Different Reference Frame Selection 17, 19

CHAPTER 5 CONCLUSION 20

CHAPTER 6 FUTURE WORK 21

CHAPTER 7. References 22

Table of Symbols

Symbol	Description
3D	Three dimensional
2D	Two dimensional
6D	Six dimensional
et al.	And others
RGBD	Red, Green, Blue, Depth
fps	Frames per second
D	2D depth map
N	Number of frames
i	Index of frame
T-SNE	t-distributed stochastic neighbor embedding
F	Set of indices of all frames ordered by ascending order of distance
d(a, b)	Distance function, returns distance between a and b
ρ	Distance threshold
R	Set of indices of all reference frames
proj(a)	Function to project 3D point to 2D
r	Index of reference frame
k	Index of joint
L	3D point
l	2D point
p(r)	Function that returns the parent joint for the given joint
z(L)	Function that returns the depth of the point

ε	Depth interval threshold
SLSQP	Sequential Least Squares Programming
SIFT	Scale Invariant Feature Transform
λ	Weights
MSRA	Microsoft Research Lab – Asia
MCP	Metacarpophalangeal
DIP	Distal interphalangeal
PIP	Proximal interphalangeal

Table 1: Table of Symbols

CHAPTER 1:

INTRODUCTION

Hand pose estimation is the task of finding the bone joints in the hand to infer the pose information from a given image or video frame. Accurate hand pose estimation is a very significant task that has many practical applications such as sign language recognition, human-computer interaction, and augmented reality applications.

Recent works on hand pose estimation methods have relied heavily on deep neural networks which require huge amount of training data [1] [2]. However, creating this training data is a difficult problem. Even with 3D sensors that use structured light or time-of-flight sensors, the problem remains a very challenging one; since human hands have a high degree of freedom and exhibits self-similarity and self-occlusion in a monocular camera setting [3].

Vision based reconstruction of the 3D pose of human hands from 2D images is really difficult since any given 2D point in the image plane can correspond to multiple 3D points in the world space, and all of these possible 3D points project into the same 2D point [4]. In addition to the other challenges related to hand pose estimation, 3D hand pose estimation from monocular 2D images also suffer from depth and scale ambiguities.

Accurate hand pose annotation has been the bottleneck for creating large-scale hand pose datasets. To avoid this problem Yuan et al. used six 6D magnetic sensors to create a million-scale dataset [5]. There are other methods such as Glauser et al. that used stretch-sensing soft hand gloves to capture the hand pose more accurately [6]. These methods require small wearable magnetic sensors to be placed at each finger joint to capture the hand pose. But the sensitivity of magnetic sensors increases with size; meaning small sensors lack in precision and are easier to be disturbed by external magnetic fields [7]. Also, magnetic sensors are not very practical to use in day-to-day operations and therefore we focus our attention on marker-less hand pose estimation methods.

Some marker-less hand pose estimation methods rely on synthetic images since it allows to create virtually infinite training data with large variations in shapes and view-points and produce annotations that

are highly accurate in case of occlusions [8]. For this reason, many synthetic image datasets were introduced in recent years [8] [9] [10] with number of frames ranging from three hundred thousand to five million frames. However, synthetic hands exhibit a certain level of deviance from real images as these do not capture the sensor characteristics such as noise and missing data that are present in real images.

There are also marker-less hand pose datasets of real hand images [11] [12] [13], that contain frames in the range of (80k-330k) but exhibit significant errors in 3D locations of the joints. We show some examples of annotation errors on the MSRA dataset [11] in Figure 1. Also, some real hand pose datasets have frames in the range of couple of thousands that are not suitable for training large neural networks.

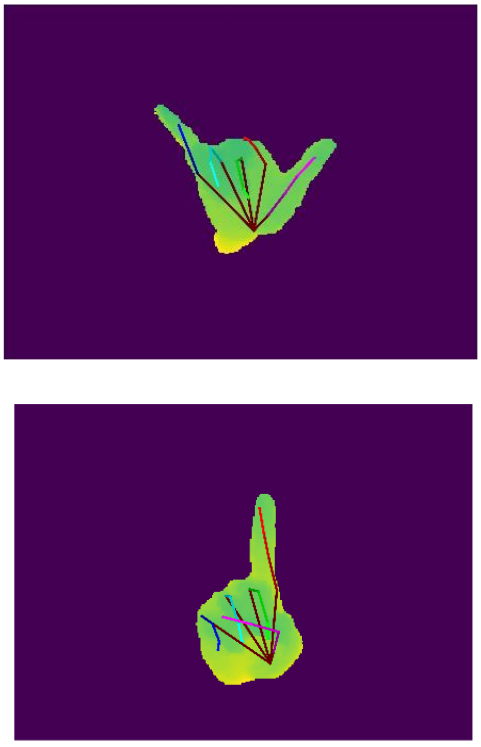


Figure 1: Annotation errors in the MSRA dataset: In the first figure the joints on the little finger and in the second image joints on the thumb and index finger aren't correctly annotated.

The method proposed in the paper annotates depth images by selecting a subset of the frames that are representative of the entire dataset, manually annotating them and passing these annotations to other frames. This method was inspired by Oberweger et al. [14], however, our major contribution to this approach was the way in which reference frames are being selected and following the same approach as

Oberweger's paper to propagate the annotations from reference frames to the rest of the frames. Instead of the original clustering method adopted by the paper we took to sorting the frames based on cosine similarity and selecting reference frames based on a threshold. The threshold is a more robust way of selecting reference frames which are representative of the entire dataset as we can compare it to the mean and median distance of the dataset which is not the case in clustering algorithms. This helps us selecting appropriate number of reference frames to get the best annotations.

CHAPTER 2:

RELATED WORK

A large body of literature is devoted to real-time pose estimation of marker less articulatable objects, such as human bodies, hands, and other man-made objects. As the primary contribution of our work is based on human hand pose estimation, we will mainly discuss the most relevant prior work on hand pose estimation.

Inferring the pose of hands is difficult problem because of self-occlusion and to mitigate this problem, Johnathan Thompson et al. [13] used 3 RGBD cameras at viewpoints separated by approximately 45˚s surrounding the user from the front. They used a predefined 3D hand model that was manually readjusted for poses that failed to fit correctly. Also, the dataset was collected from the frontal view of the user, limiting the range of poses. Simon et al. [1] proposed multi-view bootstrapping based on the observation that even if a particular image of a hand has significant occlusion from one view, there often exists an un-occluded view to recover the pose. They used 31 high definition cameras to localize subsets of key points in good views and used 3D triangulation to filter out incorrect detections. Ballan et al. [15] used eight simultaneous cameras recording at 50 fps to capture hand interactions. However, these methods impose a lot of restrictions which limit the range of hand poses.

Several other groups have used single camera setups for hand pose estimation. For example, Tang et al. [12] used a latent tree model to reflect the hierarchical topology of the hand and used it for 3D hand pose estimation from a single depth image. Although the method is limited to one hand and do not work for interaction with the other hand or objects. Sun et al. [11] used a single depth camera to fit a predefined 3D hand model using hand pose regression, which starts by initializing a rough estimate of the hand pose and then iteratively estimates the palm joints and finally figures out the finger joints keeping the palm joints fixed. Qian et al. [16] also proposed a sphere-based hand model and a novel cost function for real-time hand tracking based on a single depth camera. These tracking require manual supervision which leads to many errors. Also, tracking based methods are prone to errors from the previous frames as it does not reinitialize the estimation at each frame.

Annotating 3d hand joints in real depth images is a complicated and time-consuming process. Therefore, some researchers have focused on synthetic datasets [17] [18] [10] to generate highly accurate training data. Real datasets are limited in quantity and coverage due to the difficulty of annotations [5], but synthetic datasets do not have this problem. However, synthetic datasets exhibit a certain level of appearance difference with real images and tend to have hand kinematics which are impossible to generate using real human hands. They also lack the depth sensor noise present in real data. Although Xu and Cheng [19] analyzed the depth noises to minimize the negative impacts on overall performance, it is very difficult to model the sensor noise in a general way.

There are also magnetic sensor-based methods for acquiring accurate 3D locations of hand key-points. Yuan et al. [5] proposed a system with six 6D magnetic sensors and inverse kinematics to capture a million-scale benchmark dataset of real hand depth images [6] used a stretch-sensing soft glove to interactively capture hand poses in heavily occluded environments or low light conditions. However, data gloves are mostly custom-made and there is no industrial standard on the design and fabrication of such devices and are unsuitable to use in various user scenarios [7].

Rogez et al. [20] collected and annotated a benchmark dataset of real egocentric object manipulation scenes. They developed a semi-automatic labelling tool which allows to annotate partially occluded hands and fingers in 3D. A user labels the 2D location of a few joints and these are used to select the closest synthetic exemplar in the training set. A full hand pose is then created by combining the manual labelling and selected 3D exemplar and later refined manually. This process is iterated until an acceptable labelling is found. Oberweger et al. [14] extended this work by optimizing the reference frame selection and a global optimization step to enforce some shape and appearance constraints.

There have been other semi-automatic methods for annotating video sequences in Computer Vision. [21] used a semiautomatic method for resolving occlusion in augmented reality. They exploited object silhouettes in reference frames to predict the object silhouettes in the remaining frames. Wei and Chai [22] used a semi-automatic framework to model human motion with the help of physical and contact constraints. Compared this these works, we propose a semi-automatic approach for hand pose annotation, which minimizes the manual work, and produces more accurate 3D hand key point annotations.

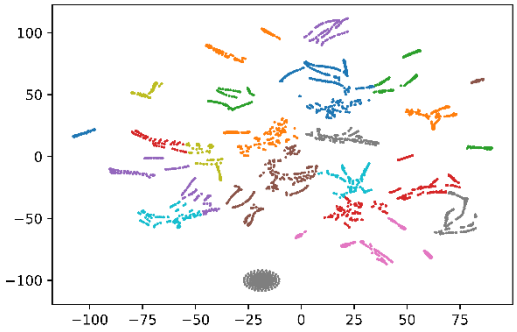
CHAPTER 3:

PROPOSED METHOD

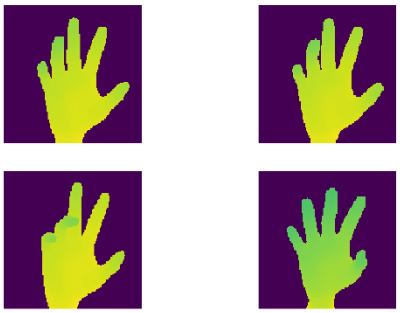
Given a sequence of depth frames $\{D_i\}_{i=0}^{N}$ capturing a hand in motion, our goal is to estimate the 3D hand joints in the depth maps while eliminating as much manual effort as possible. Our approach is based on the common observation that not all the frames in a hand depth video vary significantly from each other. Therefore, if we annotate the hand joints in a few carefully selected frames and propagate this information to similar frames it should lead to better accuracy. The process starts by automatically selecting the frames for manual annotation. 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 unannotated frames. Finally, we optimize annotations enforcing appearance, temporal and spatial constraints.

3.1 Selection of reference frame

Our goal here is to select a set of frames whose annotations will be propagated to the remaining frames in the dataset. The easiest way to achieve this would be to sample a frame after a fixed time interval; say every 10^{th} frame. However, this would not yield an ideal solution as the rate at which hand poses change in real images may vary. If the subject is slow, we might have too many reference frames increasing redundant annotations or if the subject is fast, we might lose out on key frames. Selecting these frames is crucial to our task as we want to select the right number of frames which are representative of the entire dataset while reducing manual annotations. To accomplish this, we came up with a novel approach of selecting reference frames with the help of the dataset and a threshold.



(a) T-SNE plot for Blender Dataset



(b) Visualization of frames that belong to the same T-SNE cluster

Figure 2: Visualization of similarity in the video sequence. We, therefore, select some frames from these clusters as reference frames, annotate them and use these annotations to improve the other frames that belong to the same cluster

Instead of temporal sampling, we approached the problem by sorting the frames with respect to ascending order of the distance between all frames in the dataset. The distance function can be dynamically selected however the one that worked best for us was cosine similarity. We start by annotating the first frame and pass on these annotations to every frame until the distance exceeds by a certain threshold. Once it does, we repeat the process by annotating the next frame. This ensures all similar frames have a reference frame to get their annotations from.

We use T-SNE [23] to visualize the high dimensional images in 2D space in Figure 2 a, where consecutive points in the plot represents similarity. As an example, we show the T-SNE plot for Blender dataset [11]. From the plots we project a few frames belonging to the same cluster Figure 2 b and we can see that the frames are quite similar to each other. However, we cannot use the centroid of the visualized clusters as reference frames because sometimes T-SNE moves far away points close to each other. Also,

we cannot assume the relative size of clusters from the T-SNE plot as T-SNE tends to expand dense clusters and shrink sparse ones. However, T-SNE confirms our assumption that some of the frames are indeed like each other.

To sort the frames in increasing order of their distance (cosine similarity) we use Equation 1.

$$F_i = \forall i \; argmin\Big(d\big(D_i, \forall j D_j\big)\Big) s.t. i \neq j$$

$$d(D_i, \forall j D_j) = \cos(\theta) = \frac{D_i . D_j}{\|D_i\| . \|D_j\|}$$

Equation 1

Here, F is a set of all frames in increasing order of distance, D_i and D_j are two depth maps and $d(D_i, \forall j D_j)$ returns the distance of all frames with respect to frame D_i. As we now have calculated the set F we use this to select the reference frames. We start by first setting frame 0 as a reference frame and iterate over the entire set. Every time the distance of the next frame is greater than a threshold ρ the next frame is considered as the reference frame. This is illustrated in the equation below.

$$R = \begin{cases} 1, & d(D_{F(i-1)}, D_{F(i)}) > \rho \\ 0, & d(D_{F(i-1)}, D_{F(i)}) \leq 0 \end{cases} i = 1 \dots n$$

Equation 2

Here, $D_{F(i-1)}$ is the previous reference frame and $D_{F(i)}$ is the current frame being compared from the sorted set of frames. 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 experimentally show that we achieve better results than the greedy reference frame selection method of Oberweger et al. [14] as our method does not cluster the data but sorts them based on their similarity. Sorting the frame makes sure that the closest frames always grouped together. 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 subpar 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. el [14] 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 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 [14].

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

$$s.t.\ \forall k \left\| L_{r,k} - L_{r,p(k)} \right\|_2^2 = d_{k,p(k)}^2$$

$$\forall k\ v_{r,k} = 1 \Rightarrow D_r[l_{r,k}] < z(L_{r,k} < D_r[L_{r,k}]) + \varepsilon$$

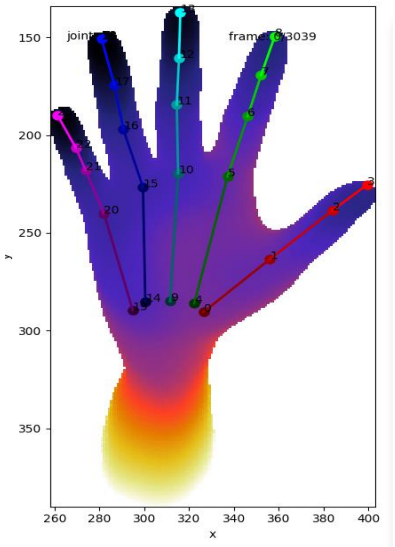
$$\forall k\ v_{r,k} = 1 \Rightarrow (L_{r,k} - L_{r,(k)})^T . c_{r,k} > 0$$

$$\forall k\ v_{r,k} = 0 \Rightarrow z(L_{r,k} > D_r[L_{r,k}])$$

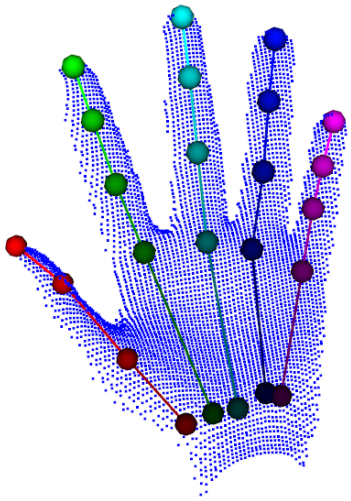
Equation 3

Where, r is the index of the reference frame. $v_{r,k} = 1$ if the k^{th} joint is visible in the r^{th} frame. $L_{r,k}$ is the 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 its 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 practice, we use ε = 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 its 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 its parent in the frame.



(a) A human annotator marks the 2D hand joint locations in the depth



(b) Visualization of the 3D hand key-points based on the 2D annotations

Figure 3: Initialization of the 3D hand joint locations in reference frames

The constraints in Equation 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 $\in$ distance of observed depth maps; (3) the z-axis value for hidden joints is greater than visible joints, and (4) depth order constraints between a joint and its parent is also maintained. We assume the lengths $d_{k,p(k)}$ are known. During implementation, we calculate this distance as the Euclidean distance between joints in the hand skeleton tree. We use SLSQP [24] to solve this problem and find the 3D hand key points i.e. $L_{r,k}$ values. Finding the hand key points 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.

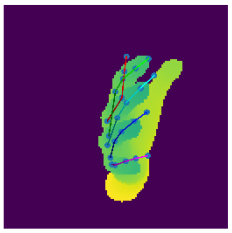
3.3 Initializing the 3D Joint Locations in the Remaining Frames

Now that we have the 3D annotations for the reference frames our next step is to propagate this information to the remaining frames. Unlike Oberweger et al. [14] method of adding newly annotated frames

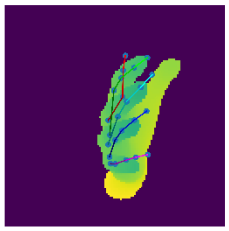
to the set of reference frame we iterate through the set F. If F_i is a reference frame skip else annotate the frame with respect to the previous frame F_{i-1}. This helps in two folds, first being the previous frame will always be annotated hence assuring that we always use the closest frame as reference frame. This was not the case in the greedy approach as the closest frame might not have been annotated yet. Secondly, we eliminate the process of recalculating the nearest reference frame saving on computation time.

To align the nearest frame we follow the method suggested by Oberweger et al [14], using SIFT-Flow [25]. Unlike Optical flow which aligns an image to its temporally adjacent frame, SIFT-Flow aligns an image to its 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.

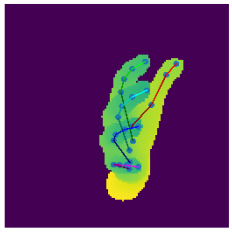
We initialize SIFT-Flow by aligning the closest reference frame F_{i-1} to a non-reference frame F_i. This maps the 2D reprojection of joints in frame F_{i-1} to 2D locations in frame.F_i We back-project each of these 2D joint locations on the Depth map D_c to initialize $L_{F(i),k}$. We check for each 3D joint in the non-reference frame i.e. $L_{F(i),k}$, the distance to its parent joint $L_{F(i),p(k)}$ and thereby enforcing the 3D distance between joints in the hand skeleton tree. The initialization of a non-reference frame using its closest reference frame is illustrated in Figure 4.



(a) Closest Reference frame selected



(b) Initializing the current frame with SIFT-Flow



(c) Result after optimization

Figure 4: SIFTFlow optimization of the remaining frames based on the closest reference frame

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. [14]:

$$\sum_{i \in [1;N]} \sum_{k} ds\left(D_i, proj\left(L_{i,k}\right); D_i, l_{i,j}\right)^2 + \quad \text{(C)}$$

$$\lambda_M \sum_{i} \sum_{k} \left\| L_{i,k} - L_{i+1,k} \right\|_2^2 + \quad \text{(TC)}$$

$$\lambda_P \sum_{r \in R} \sum_{k} v_{r,k} \left\| proj(L_{r,k} - l_{r,k}) \right\|_2^2 \quad \text{(P)}$$

$$s.t. \forall i,k \left\| L_{i,k} - L_{i,p(k)} \right\|_2^2 = d_{k,p(k)}^2$$

Equation 4

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 cannot 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.

As the remaining frames were optimized with the closest reference frames in the previous step, we allowed the joint positions to change without enforcing the shape or temporal constraints. In this step, 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:

EVALUATIONS

We apply our method on both synthetic and real hand pose datasets and compare our results with the results of Oberweger et al. [14]. We test our novel reference frame selection method on a synthetic dataset called Blender created by [14] and achieve better results. We provide quantitative comparison of our results in the next section. Later we show our results are also superior on a real hand pose dataset (MSRA [11]) and show the qualitative comparisons as well. Finally, we show that our reference frame selection is more flexible as users can choose either higher or lower number of reference frames for annotation. We also show the quantitative results for different number of reference frame selection.

4.1 Evaluation on Synthetic Data

We used the Blender dataset, which contains 3040 depth frames from a single camera setup to test our method. As we know the actual 3D hand joint locations for all frames in the synthetic dataset, we can use the annotations for the reference frame and infer the 3D hand joint locations in the remaining frames. Later, we can compare our inferred joints with the ground truth values to estimate the mean, median and max errors for our method by calculating the Euclidean distance between the two. In the following table, we show that our reference frame selection method based on cosine distance achieve better initialization results than Oberweger et al. [14] in the following Table 2:

	Our Method	**Oberweger et al. [14]**
Mean error	8.01	10.07
Max error	89.18	99.05
Median error	6.21	6.93

Table 2: Comparison of Initialization results on the Blender Dataset

Also, our initialization results later propagate to better overall results, which we show quantitatively in the following Table 3:

	Our Method	Oberweger et al. [14]
Mean error	4.69	4.91
Max error	84.33	73.65
Median error	3.35	3.68

Table 3: Comparison of Final Results on the Blender Dataset

4.2 Evaluation on Real Data

We also test our results on a real hand pose dataset. We choose MSRA which contains 76K depth frames from 9 different subjects and 21 hand key-points. We test our method on the first 8500 frames on the dataset. Based on our reference frame selection method, we select a subset of frames for human annotation based on the distance threshold ρ. Unlike selecting 10% of the frames as reference frames [14], we can select a higher or lower number of reference frames based on the distance threshold. After the human annotator provides the 2D hand joint locations on the selected reference frames, we can figure out the 3D hand key-points using SLSQP. Later, we align the 3D hand key-points of each of the remaining frames to their closest reference frame using SIFT-Flow and enforce the shape and appearance constraints to get the refined annotations.

Unlike synthetic hand pose datasets, we do not know the actual position of hand joints in a real hand pose dataset. Therefore, we show some qualitative results on how the method performed on the MSRA [11] dataset in the following Figure 5.

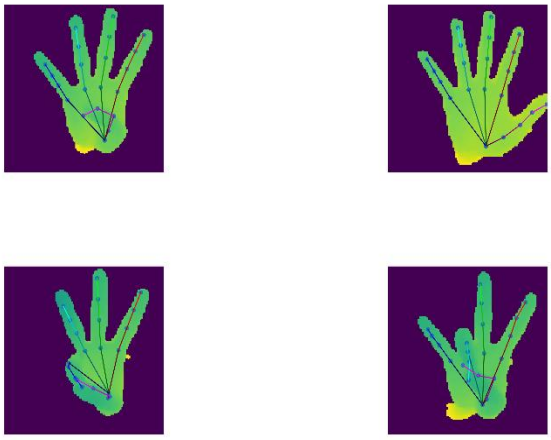


Figure 5: Qualitative Results on the MSRA dataset

4.3 Evaluation on Different Reference Frame Selection

One of the main benefits of our reference frame selection is that we can choose a dynamic no of frames based on the cosine distance value. 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 4:

	~2% frames	**~5% frames**	**~10% frames**
Mean error	5.79	5.5	4.69
Max error	76.80	78.99	84.33
Median error	4.61	4.18	3.36

Table 4: Evaluation on the Blender Dataset for different % of reference frame selection

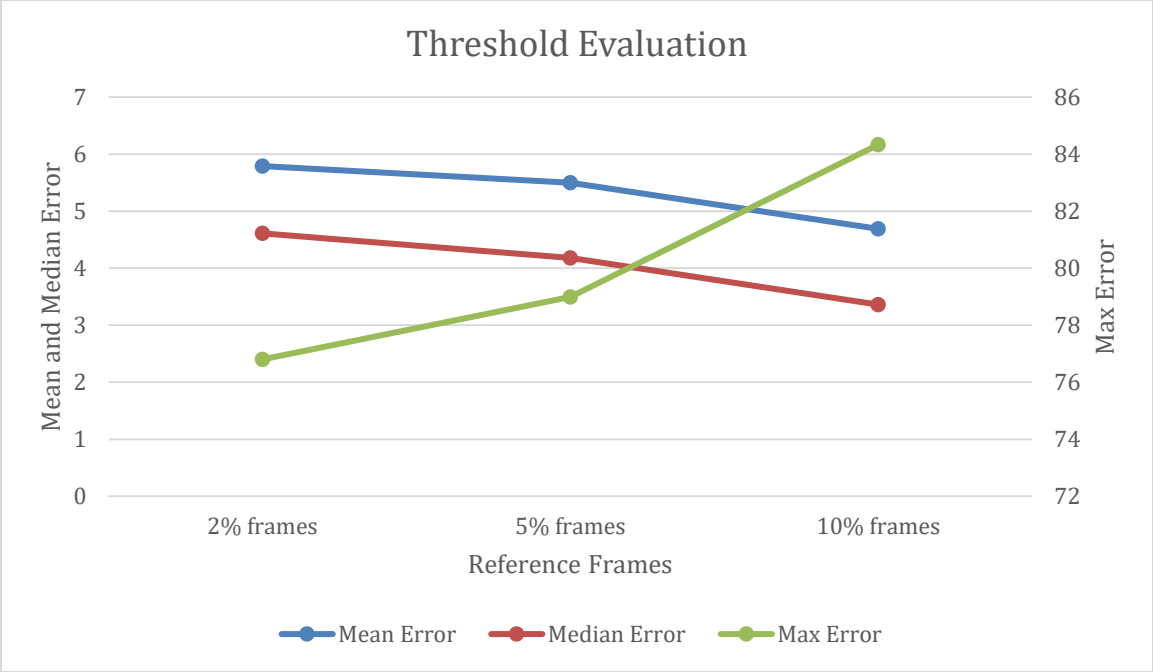


Figure 6: Graph showing errors depicted in table 4

As we can see from Figure 6 and Table 4, the annotation results improved with the selection of more reference frames, i.e. with more human annotations we can get better results. However, we got comparably good results while selecting 2-5% of the frames as reference frames. This shows that with minimal effort we can produce better annotations in the overall dataset.

To further solidify this claim we show the distribution of the selected reference frames in the T-SNE graph. This will give us a better idea of how our selection method is well distributed throughout the dataset and encompasses all the frames.

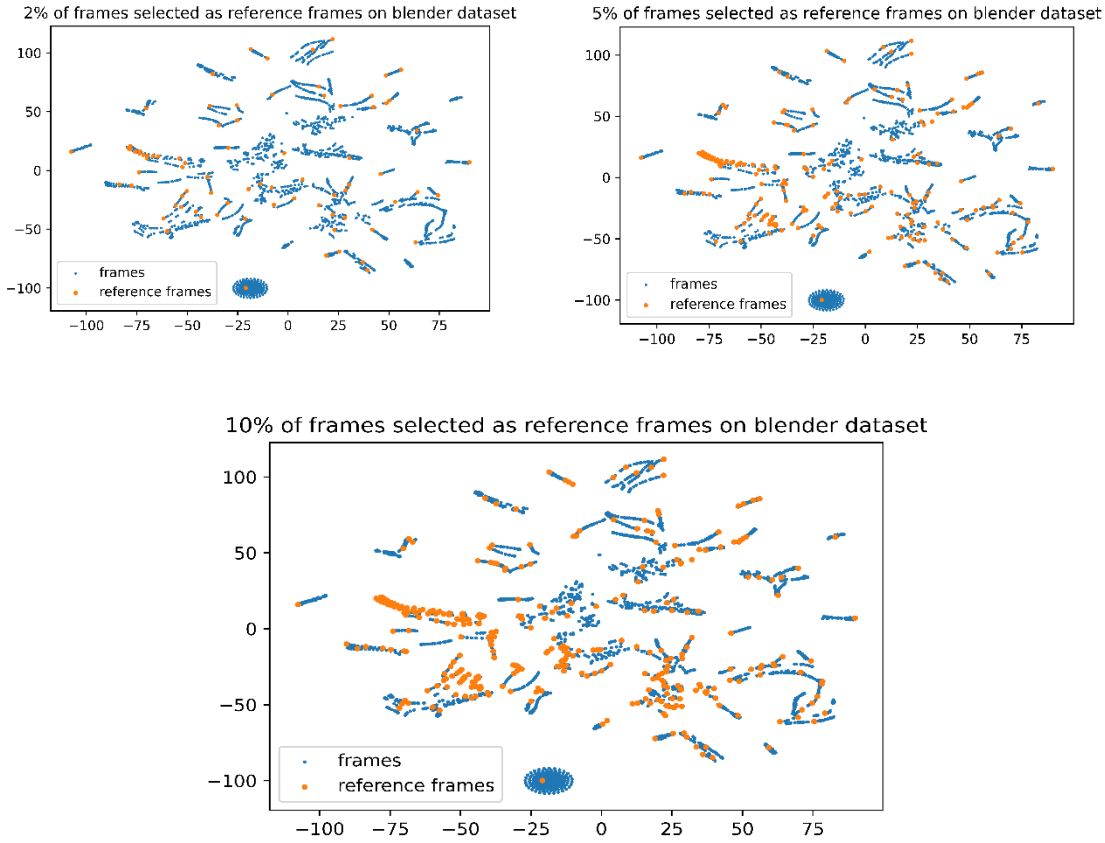


Figure 7: T-SNE distribution of reference frames with respect to non-reference frames.

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. Our method provides a solution by processing frames with some manual supervision and propagating this information to other frames to get overall better 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.

CHAPTER 6.

FUTURE WORK

There are couple of ways in which this work can be extended. First would be to create a more intuitive UI which would enable users to annotate directly in 3D. The major drawback with annotating in 2D and projecting it to 3D is having to repeat annotations multiple times to get the right annotations. This task is time consuming and erroneous. If annotations can be done in 3D by stitching in together multiple 2D view annotation.

Secondly, we can improve annotations in 3D by adding biological constraints. These constraints would make sure that the joints do not deviate from normal hand poses and can be integrated into Equation 3. [26] state the following type 1 constraints.

$$0^{\circ} \leq \theta_{MCP-F} \leq 90^{\circ}$$

$$0^{\circ} \leq \theta_{PIP} \leq 110^{\circ}$$

$$0^{\circ} \leq \theta_{DIP} \leq 90^{\circ}$$

$$-15^{\circ} \leq \theta_{MCP-AA} \leq 15^{\circ}$$

Equation 5

Where θ stands for the angle between the joints and the subscripts define the name of the joint and motion. F stands for flexion angles and AA stands for abduction/adduction.

CHAPTER 7.

REFERENCES

[1] H. J. I. M. a. Y. S. Tomas Simon, "Hand keypoint detection in single images using multiview bootstrapping.," In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017.

[2] S. Y. a. T.-K. K. Qi Ye, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," In European conference on computer vision, 2016.

[3] P. W. a. V. L. Markus Oberweger, "Hands deep in deep learning for hand pose estimation," arXiv, 2015.

[4] U. I. P. M. O. H. a. J. K. Adrian Spurr, "Weakly supervised 3d hand pose estimation via biomechanical constraints.," arXiv, 2020.

[5] Q. Y. B. S. S. J. a. T.-K. K. Shanxin Yuan, "Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis.," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[6] S. W. D. P. O. H. a. O. S.-H. Oliver Glauser, "Interactive hand pose estimation using a stretch-sensing soft glove," ACM Transactions on Graphics, 2019.

[7] C. Y. C. T. Z. L. J. T. S. O. Y. F. a. Z. X. Weiya Chen, "A survey on hand pose estimation with wearable sensors and computer-vision-based methods.," Sensors, 2020.

[8] A. E. F. N. K. V. K. T. A. H. a. D. S. Jameel Malik, "Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth.," International Conference on 3D Vision, 2018.

[9] F. B. O. S. D. M. S. S. D. C. a. C. T. Franziska Mueller, "Ganerated hands for real-time 3d hand tracking from monocular rgb," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[10] D. M. O. S. S. S. D. C. a. C. T. Franziska Mueller, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017.

[11] Y. S. L. X. T. a. J. S. Xiao Sun, "Cascaded hand pose regression," In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[12] H. J. C. A. T. a. T.-K. K. Danhang Tang, "Latent regression forest: Structured estimation of 3d articulated hand posture," In Proceedings of the IEEE conference and pattern recognition, 2014.

[13] M. S. Y. L. a. K. P. Jonathan Tompson, "Real-time continuous pose recovery of human hands using convolutional networks," ACM Transactions on Graphics, 2014.

[14] G. R. P. a. V. L. Markus Oberweger, "Efficiently creating 3d training data for fine hand pose estimation.," In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[15] A. T. J. G. L. V. G. a. M. P. Luca Ballan, "Motion capture of hands in action using," European Conference on Computer Vision, 2012.

[16] X. S. Y. W. X. T. a. J. S. Chen Qian, "Realtime and robust hand tracking from depth," In Proceedings. of the IEEE conference on computer vision and pattern recognition, 2014.

[17] C. W. a. T. M. Fanqing Lin, "Two-hand global 3d pose estimation using monocular rgb," arXiv, 2006.

[18] C. Z. a. T. Brox., "Learning to estimate 3d hand pose from single rgb images.," In Proceedings of the IEEE international conference on computer vision, 2017.

[19] C. X. a. L. Cheng., "Efficient hand pose estimation from a single depth image," In Proceedings of the IEEE international conference on computer vision, 2013.

[20] M. K. J. S. I. J. M. M. M. a. D. R. Gregory Rogez, "3D Hand Pose Detection in Egocentric RGB-D Images," In European Conference on Computer Vision, 2014.

[21] V. L. a. M.-O. Berger, "A semi-automatic method for resolving occlusion in augmented reality," In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[22] X. W. a. J. Chai., "Videomocap: Modeling physically realistic human motion from monocular video sequences," In ACM SIGGRAPH, 2010.

[23] L. v. d. M. a. G. Hinton, "Visualizing data using t-sne.," Journal of machine learning research, 2008.

[24] D. K. e. al., "A software package for sequential quadratic programming.," 1988.

[25] J. Y. a. A. T. Ce Liu, "Sift flow: Dense correspondence across scenes and its applications.," IEEE transactions on pattern analysis and machine intelligence,, 2010.

[26] Y. W. T. S. H. John Lin, "Modeling the Constraints of Human Hand Motion," IEEE, Austin, 2000.