UNIVERSITY OF CALIFORNIA
RIVERSIDE

Human Body Recognition on 3D Mesh Using Local Surface Patch

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Nghia Nguyen

March 2024

Thesis Committee:

Dr. Bir Bhanu, Chairperson
Dr. Greg Ver Steeg
Dr. Daniel Wong

Acknowledgments

I am deeply indebted to my thesis advisor, Prof. Bir Bhanu, whose unwavering support and insightful suggestions were pivotal in the refinement of my ideas and the successful completion of my thesis. Prof. Bhanu's ability to guide me through various challenges with innovative solutions and encouragement was instrumental in my journey. His mentorship extended beyond mere academic advice, providing me with the resilience and inspiration needed to persevere through the toughest moments of my academic endeavor.

I would also like to express my sincere gratitude to my thesis committee members, Prof. Greg Ver Steeg and Prof. Daniel Wong. Their exceptional teachings in their respective courses laid the foundation for my academic interests and research direction. Beyond the classroom, their continued support and advice played a crucial role in my development as a researcher. Their willingness to contribute their time and expertise to my thesis, even amidst their busy schedules, has left a lasting impact on me.

My heartfelt thanks go to all the members of the VISLAB at the University of California, Riverside. The seminars and discussions held by this group exposed me to cutting-edge research and developments in our field. This environment of intellectual exchange and camaraderie has been invaluable in broadening my understanding and fostering a deep appreciation for innovative research.

I am also grateful to my graduate advisors, Vanda Yamaguchi, Prof. Marek Chrobak, and Marisa Mendoza, for their guidance through the myriad administrative processes and academic inquiries during my time at UCR. Despite numerous procedural errors and queries on my part, their patience and assistance ensured that I remained on track

towards completing my thesis. Their support was a beacon of light during moments of uncertainty and complexity.

Lastly, but most importantly, I wish to extend my deepest gratitude to my family—my brother and my parents. Their financial and emotional support has been the bedrock of my journey. The sacrifices they have made and their unwavering belief in my abilities have fueled my ambition and sustained me through challenging times. Without their love and encouragement, reaching this milestone would have been unimaginable.

In conclusion, the collective wisdom, encouragement, and support from these remarkable individuals have shaped my academic journey in profound ways. I am forever grateful for their contributions to my growth, both professionally and personally.

To my family for all the support.

ABSTRACT OF THE THESIS

Human Body Recognition on 3D Mesh Using Local Surface Patch

by

Nghia Nguyen

Master of Science, Graduate Program in Computer Science
University of California, Riverside, March 2024
Dr. Bir Bhanu, Chairperson

Most human recognition methods rely on unique bio-metric features like facial structure, fingerprints, iris, voice, hand, or gait. However, human body shape also offers a distinctive metric for identification and is very important for various applications, including crime prevention, forensic identification, and security monitoring, especially when the face of a person cannot captured by camera. Recently, with the raising of 3D objects in today's technology landscape, especially in virtual reality (VR) and augmented reality (AR), people have created and enhanced many methods to regress accurate 3D human bodies from one or many RGB images. These methods are really helpful not only in the field of creating 3D objects in AR or VR but also in human recognition. Many human recognition systems recently integrated these 3D object regression methods in their neural network to make their deep learning models understand more about humans in 3D. This raises the question of, whether these 3D objects are accurate enough to recognize humans. Almost no research has tried to recognize humans on 3D objects. Our research aims to figure out whether these 3D human bodies regressed from 2D images from the state-of-the-art (SOTA) method are good enough to recognize humans. We used SHAPY, a SOTA model for generating a 3D human body mesh from an RGB image. While deep learning has recently shown notable results in processing directly point cloud to do some tasks related to 3D objects like 3D object classification, or segmentation, it is very hard to explain the results of deep learning methods. Therefore we are using Local Surface Patch (LSP) which is a geometric way to extract shape features from 3D meshes. Local Surface Patch makes it very easy to verify if recognition in 3D mesh from SHAPY works, and if it doesn't work, we can explain why. We have done a lot of experiments to show that 3D meshes created from the state-of-the-art method are currently not good enough for human body recognition. We also implement and improve LSP methods for 3D objects and show that it is robust to capture shape features from 3D objects. We also propose a solution to alleviate the effect of high variance 3D body shape from SHAPY and improve the recognition results.

Keywords: 3D Human Body Recognition, 3D Human Pose and Shape, 3D Object Recognition, SHAPY, Local Surface Patch

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human recognition aims to identify or verify the identity of a person based on their physiological or behavioral characteristics. Most human recognition methods rely on unique bio-metric features like facial structure, fingerprints, iris, voice, hand, or gait. However, human body shape also offers a distinctive metric for identification, and very important for various applications, including crime prevention, forensic identification and security monitoring, especially when face of person cannot captured by camera.

Recently, with the raising of 3D objects in today's technology landscape, especially in virtual reality and augmented reality, people are creating and enhancing many methods to regress accurate 3D human bodies from one or many RGB images. These methods are really helpful not only in the field of creating 3D objects in AR or VR but also in human recognition. Many human recognition systems recently integrated these 3D object regression methods in their neural network to make their deep learning models understand more about humans in 3D. This raises the question of, whether these 3D objects are accurate enough to recognize humans. Almost no research has tried to recognize humans on 3D objects. Our research aims to figure out whether these 3D human bodies regressed from RGB images are good enough to recognize humans. Our research opens a way to recognize the human body on 3D objects, which in the future we can get even more accurate 3D human body based on 3D scanners or more accurate methods later on.

We utilize SHAPY, a state-of-the-art (SOTA) model for generating a 3D human body mesh from an RGB image. SHAPY shows that it has very good results in regressing accurate body shape, compared to previous SOTA methods. After we get 3D human bodies from SHAPY, we will evaluate whether these SHAPY 3D meshes are good enough for human body recognition. While deep learning has recently shown notable results in processing directly point cloud to do some tasks related to 3D objects like 3D object classification, or segmentation, it is very hard to explain the results of deep learning methods. Because our research focuses on whether these 3D meshes are good enough, so it's uncertain whether these meshes can be used for human recognition. Therefore, we are using Local Surface Patch (LSP) which is a geometric way to extract shape features from 3D meshes. Local Surface Patch makes it very easy to verify if recognition in 3D mesh from SHAPY works, and if it doesn't work, we can explain why. Then, we also indexed and retrieved based on the Local Surface Patch. We have done a lot of experiments to show our results and our conclusion. We also implement a deep learning-based approach to see what kind of results we can have.

In summary, our contributions are:

- We do multiple experiments and visualize our results to conclude that 3D meshes created from the state-of-the-art method currently can not be used for human body recognition.

- We improve LSP methods, make them work with 3D objects, enhance the LSP method on 3D objects, and do multiple experiments to show that it is robust to capture shape features from 3D objects.
- We also propose a solution to alleviate the effect of high variance 3D body shape from the state-of-the-art method and improve the recognition results with our approach.

Through our research, we aim to broaden the scope of human recognition by expanding beyond the traditional focus on bio-metric features and images or video to include the 3D human body.

Chapter 2

Related Works

In this chapter, we discuss about two 3D data representations including point cloud and mesh in Section 2.1. In Section 2.2, we discuss about many 3D human body models out there and their histories of development. Then we discuss about the overview and classification of the 3D Human Pose and Shape field, and list some notable methods for regressing 3D human mesh from a single RGB image using 3D human body models above in Section 2.3. Then we discuss briefly about some human recognition systems recently using 3D body reconstruction to make their model understand more about humans in 3D in Section 2.4. Finally, we discuss about 3D object recognition methods, including local surface patches and deep learning in Section 2.5.

2.1 3D Data Representation

There are two common 3D data representations: **Point Cloud** and **Mesh** as you can see in Figure 2.1.

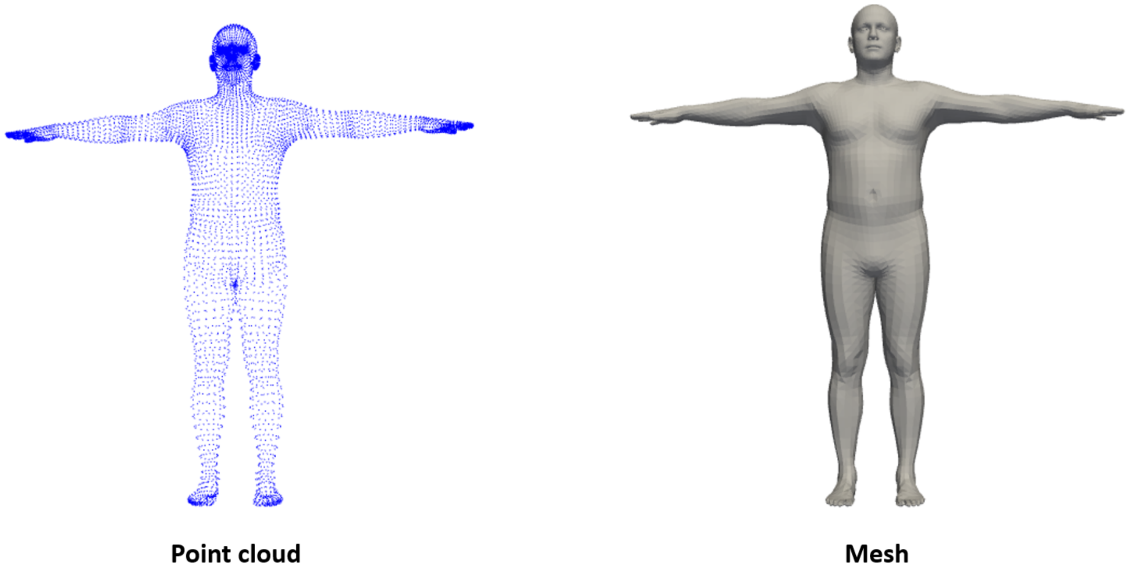


Figure 2.1: 3D data representation: Point cloud and mesh

- **Point cloud:** A Point Cloud is a collection of points in a 3D coordinate system where each point represents a tiny part of an object's surface. Point cloud representation doesn't include information about how points are connected or how the surface continuity is. Recent deep learning methods obtain a point cloud from an object and directly process it in a neural network model.

- **Mesh:** A Mesh is a collection of vertices, edges, and faces that form a 3D object. The vertices are points in 3D space, the edges connect pairs of vertices, while the faces are the two-dimensional polygons that enclose the edges to form the surface of the object. All 3D human body models in Section **??** generate 3D mesh representation. Because a mesh has a relationship between vertices, it facilitates the computation of geometric features from a 3D mesh.

2.2 3D Human Body Model

3D human body models are models that create a 3D mesh from parameters. These parameters can be shape parameters, pose parameters, hand parameters, face parameters, ... In this section, we describe briefly many 3D human body models, their novelty, and how each of them is improved upon its predecessors.

2.2.1 SCAPE (Shape Completion and Animation of PEople)

In 2005, SCAPE [1] led the way for 3D human body modeling by separately modeling body shape and pose deformations. This approach also allows for capturing muscle and skeleton movements. Therefore, it can express the human body in various poses. However, due to its complexity, it's limited to real-time applications.

2.2.2 SMPL (Skinned Multi-Person Linear Model)

In 2015, SMPL [19] was shown to prove that it improved upon SCAPE. SMPL simplified and enhanced the modeling process by using linear blend skinning with corrective blend shapes. Therefore, it allows efficient and realistic modeling of pose-dependent deformations. Its parameterization of body shapes and poses enabled easier manipulation (providing control for re-shaping and re-posing) and broader application, including real-time animations.

However, the SMPL 3D model has some limitations. Firstly, SMPL sometimes gets confused and thinks parts of the body that are far apart (like arms and toes) should move together. Secondly, SMPL doesn't pay enough attention to how someone's shape (like how muscular or slender they are) changes the way they move. Thirdly, When we move, our muscles and skin shift in complex ways. SMPL tries to mimic this with a simple method, which doesn't always capture the real-life wiggles and jiggles of our bodies. Researchers noticed these problems and have been working on upgrades like STAR [21] which is being smarter about which parts of the body should move together and which shouldn't, or SoftSMPL [26] which makes the movement of muscles and skin more realistic.

2.2.3 Adam

In 2018, Adam [12] improved upon SMLP by capturing multiple scales of human movement, including facial expression, body motion, and hand gestures. Adam also can express hair and clothing geometry, making it possible to fit people in daily life scenes. Therefore, it can be used for motion tracking, capturing body movements, facial expressions, and hand motion simultaneously.

2.2.4 SMPL-X

SMPL-X [22] was introduced in 2019 as an extension of SMPL. It incorporates highly detailed facial expressions (FLAME head model) and finger movements (based on the MANO model) into the SMPL model. SMPL-X significantly enhanced the model's expressiveness and realism for generating 3D human body mesh.

SMPL-X has totally 119 parameters representing the body, hand and face:

- 75 parameters for the global body rotation and body, eyes , jaw joints.

- 24 parameters for the lower dimensional hand pose PCA space.

- 10 parameters for the body shape.

- 10 for the facial expressions.

In the next sections, we describe several methods that utilize these 3D body models to regress the 3D human body from a single RGB image.

2.3 3D Human Pose and Shape (HPS)

3D Human Pose and Shape (HPS) are methods that create 3D human bodies from one or more RGB images. These methods often generate 3D model parameters (shape, pose, ...) from RGB images using deep neural networks and then generate 3D mesh from these parameters. These methods can be divided into two broad categories: **parametric methods** and **non-parametric methods**.

Parametric methods generate parameters to input into a 3D body model, such as SCAPE [1], SMPL [19], Adam [12], SMPL-X [22] and GHUM [34]. Parametric methods are divided into two categories: optimization-based methods and regression-based methods. Optimization-based methods [2, 3, 9, 22] are methods that iterative refine parameters (which are input into a 3D body model) during training in order to improve and optimize these parameters. For example, these methods can initialize these parameters, render a 3D human body from these parameters using a 3D body model, get 3D joint locations, project these 3D joint locations to get 2D joint locations and compare with ground truth 2D joint locations to refine these parameters. On the other hand, regression-based methods [7, 8, 11, 13, 14, 15, 17, 20, 35] are based on deep neural networks to directly regress these parameters from RGB images.

Non-parametric methods predict 3D body models directly [10, 25, 30, 33], so these methods can generate clothing or hair that are not available in parametric methods.

2.3.1 Human Mesh Recovery (HMR)

Human Mesh Recovery (HMR) [13] is an end-to-end framework for reconstructing a full 3D mesh (SMPL model) of a human body from a single RGB image. Unlike another method that first predicts 2D joint location and then predicts 3D join locations, HMR infers 3D pose and shape parameters directly from image pixels. Its novelty lies in minimizing the re-projection loss of key points and introducing an adversary trained to verify whether human body shape and pose parameters are real or not using a 3D human meshes large database.

2.3.2 SMPL oPtimization IN the loop (SPIN)

SPIN (SMPL oPtimization IN the loop) [15] is an approach that combines regression-based method and optimization-based method. This method initializes its iterative optimization by directly regressing 3D pose and shape parameters from image pixels using HMR, then from those initial parameters, it repeat the optimization process by fitting the body model to 2D joints within the training loop. In short, SPIN is self-improving by nature, training is feasible even when no image with 3D ground truth is available, and it outperforms the results of HMR. SPIN also used the SMPL model.

2.3.3 Synthetic Training for Real Accurate Pose and Shape (STRAPS)

STRAPS [28] addresses the issue of inaccurate body shape due to lacking in-the-wild training data by using silhouettes and 2D joints as inputs to a shape and pose regression neural network, and trained with synthetic training data by sampling SMPL pose and shape parameters and decoding them into 3D vertices and joints, which are projected, rendered and corrupted to form an input proxy representation. Its paper also provides a new dataset, Sports Shape and Pose 3D (SSP-3D), which contains RGB images of tightly clothed sports persons with a variety of body shapes. STRAPS shows that it has the best results on the SSP-3D dataset compared to many state-of-the-art methods, including HMR and SPIN.

2.3.4 SHAPY

SHAPY [6] also addresses the inaccurate body shape issue due to the lack of label 3D dataset by uses two different ways to collect annotations for 3D body shape: (1) collect anthropometric measurements (height, weight, . . .) from online model-agency data (2) annotate images with linguistic shape attributes (long neck: 3/5, soft body: 4/5) using crowd-sourcing. SHAPY learns the mappings between the body shape, body measurements, and linguistic attributes: (1) virtual measurements (body shape to measurements); (2) attributes to shape; (3) shape to attributes. These mappings make the model learning without having explicitly labeled datasets. SHAPY paper also provides a new dataset, Human Bodies in the Wild (HBW). SHAPY shows that it has the best results on HBW, MMTS, and SSP-3D datasets compared to any state-of-the-art method before, including HMR, SPIN, and STRAPS. SHAPY used SMPL-X model.

2.4 Human recognition using 3D body reconstruction

Previously, many human recognition or human re-identification approaches tried to recognize humans by processing RGB images and ignoring the prior that the human body is a 3D non-rigid object. Recently research [29, 18] tried to incorporate 3D body reconstruction into their pipelines or their model architectures to make their models understand more about humans in 3D, then make it more robust to obtain accurate shape features.

For example, HMID [29] use HMR to generate 3D SMPL mesh and integrate Chamfer loss (compute difference between 3D mesh point cloud and points in silhouette cloud) in their training pipeline to make the model produce more accurate shape and pose parameters. 3DInvarReID [18] proposes a novel long-term re-identification method that learns shape, and cloth in 3D, then uses a neural linear blend skin to generate 3D human with cloth, projects to get RGB rendered human body from 3D body, and then computes the loss with the input. ShARc [36] gets the 3D shape parameters and uses them in the Pose and Shape Encoder to make the model understand more about the shape of humans in 3D. Even in gait recognition, people use human shapes in 3D to get more accurate results. GaitVIBE [27] uses shape and pose parameters to create gait embedding using gait recognition head, and then use that embedding for recognition.

Understanding humans in 3D is now really helpful for many research to achieve higher results. All of the research we discussed either used shape and pose parameters of the SMPL (or SMPL-X) model, or used 3D SMPL mesh to generate something for calculating the loss. This raises the question of what is the quality of 3D objects from the state-of-the-art method. On the way of seeking the answer to this question, our research is focused

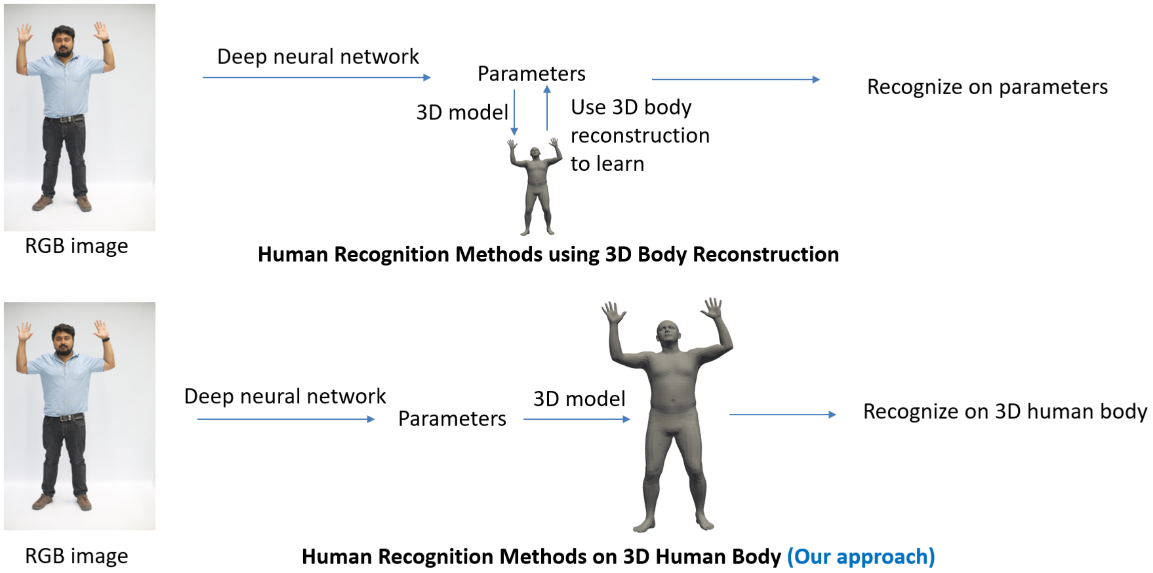

Deep neural network
Parameters
3D model
Use 3D body reconstruction to learn
Recognize on parameters
RGB image
Human Recognition Methods using 3D Body Reconstruction
Deep neural network
Parameters
3D model
Recognize on 3D human body
RGB image
Human Recognition Methods on 3D Human Body (Our approach)

Figure 2.2: Our approach compared to human recognition methods using 3D body reconstruction

on recognizing humans on 3D objects. Given the 3D human body of the state-of-the-art method, can we recognize humans with these 3D objects? Figure 2.2 compares the flow of human recognition methods using 3D body reconstruction, and human recognition methods on the 3D human body (our approach).

The next section discusses related works in 3D object recognition. We discuss about Local Surface Patch, which recognizes 3D objects using geometry calculation, and the deep learning approach, which recently has notable results in 3D object classification and segmentation.

2.5 3D Object Recognition

2.5.1 Local Surface Patch (LSP)

Definition

Local Surface Patch (LSP) [4, 5] is a region consist a feature point P and its neighbors N. LSP representation includes its surface type T_p, centroid of the patch and a histogram of shape index values vs. dot product of the surface normal at the feature point P and its neighbors N.

LSP capture the local shape feature of each point in an object. LSP has been previously shown that it has very good results in 3D object classification [4] and human ear recognition [5].

Robustness and rotation in-variance of LSP representation

With two ear images of the same person but different viewpoints, two LSPs belonging to the same physical location of these two images have similar histograms, while two LSPs of different locations on two ear images of the same person, or even two LSPs of the same location on two ear images of two different people, they have different histograms. This shows that LSP is robust and rotation in-variance and therefore, it's very powerful for recognition.

2.5.2 Deep Learning

Recently, when deep learning has become standard in computer vision tasks, people also try using deep learning on 3D object recognition, classification, or segmentation tasks.

PointNet [23] is the first deep learning model that directly processes point cloud and performs 3D object classification and segmentation tasks. It achieves high results (89% accuracy) on the public dataset ModelNet40 [32], which is a 3D dataset that contains 3D objects of 40 different classes. Many deep learning approaches later are built upon PointNet, such as PointNet++ [24], PointCNN [16], ...

While deep learning has recently shown notable results in processing directly point cloud to do some tasks related to 3D objects like 3D object classification, or segmentation, it is very hard to explain the results of deep learning methods. Because our research focuses on whether these 3D meshes are good enough, so it's uncertain whether these meshes can be used for human recognition. Therefore, we are using Local Surface Patch (LSP) which is a geometric way to extract shape features from 3D meshes. Local Surface Patch makes it very easy to verify if recognition in 3D mesh from SHAPY works, and if it doesn't work, we can explain why.

In the next chapter, we discuss about our method for using LSP to recognize the 3D human body from SHAPY.

Chapter 3

Human Body Recognition on SHAPY 3D mesh using LSP

Our approach is in Figure 3.1. First, from RGB image we use SHAPY to get the 3D human body in canonical pose. Then we obtain vertices from that 3D mesh and get LSP for each point at feature indices only. Then for each LSP, we get the kernel density estimation (KDE) from shape indices and cosine angles of all neighbor points. We then normalize it and use this vector to index and query using the vector nearest search.

Section 3.2 discusses why we use 3D mesh in the canonical pose for human recognition. Section 3.3 discusses about how we select feature indices. Section 3.4 discusses some differences change changes in computing LSP between our implementation and LSP paper [4], and then details of our LSP implementation.

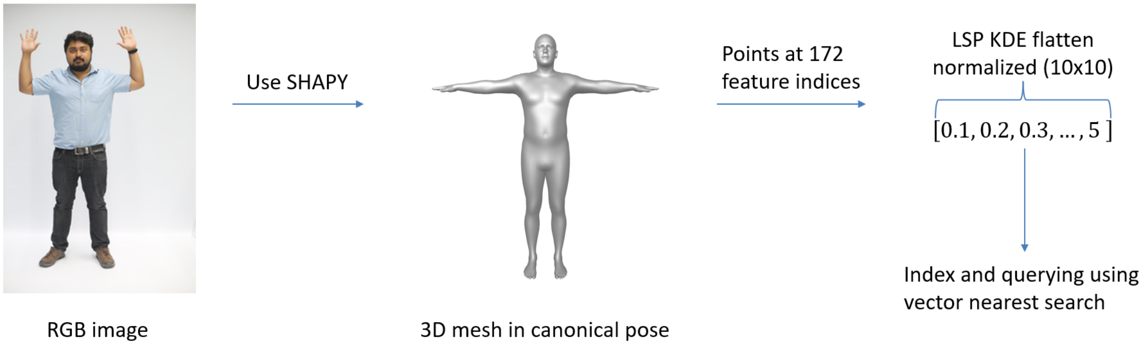


Figure 3.1: Using LSP to recognize 3d human body from SHAPY

3.1 Problem formulation

Given 3D meshes as input, can we design a solution to detect whether they are from the same or different person? Our solution is we generate 3D meshes from easy-captured RGB images using SHAPY and use LSP on generated 3D meshes for recognizing humans. Given a 3D mesh from SHAPY, we need a way to get a feature vector from each point in this mesh. Then, we will index feature vectors on gallery meshes in a database. With each query mesh, we will also get feature vectors from these meshes and then retrieve the nearest vectors to these feature vectors in that database.

The order of points in the SHAPY mesh is always preserved in terms of physical location. It means point index i^{th} at mesh A always has the same physical location as point index i^{th} at mesh B. As you can see in Figure 3.2, point index 1000 at two different mesh points at the same location, and this applies to all points in a 3D mesh from SHAPY. So given a mesh M_1 of person P which is obtained from image I_1 in the gallery and a mesh M_2 of the same person P but obtained from image I_2 in the query (probe), our mission is

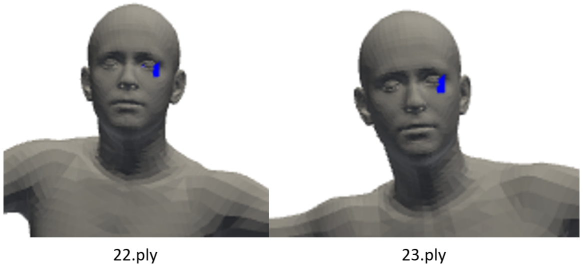


Figure 3.2: Point index 1000 (total 10475 points)

to try to match points at the same indices of mesh M_1 and mesh M_2 together.

3.2 Get SHAPY 3D mesh in canonical pose

From an RGB image, we get 3D mesh using SHAPY. Previously, we used posed mesh to recognize humans. After many experiments, we see that when matching 3D posed mesh, in a lot of cases, point index i^{th} at mesh M1 cannot match point index i^{th} at mesh M2 because of the different poses in these two meshes. Also, a lot of research performs human recognition only on the shape feature, which is the input in the SMPL-X model to form the shape for the 3D mesh. Therefore, we decided to recognize humans only on canonical pose (standard pose, T-pose), where it only retains the shape parameters but not the pose parameters. Also, this makes sense because the pose parameters are only for making the pose of 3D mesh like the pose of a person in an RGB image, it has no meaning in human recognition.

3.3 Filtering only distinctive points in mesh

Previously, we used all vertices in a mesh for indexing and querying. This made our experiment can get precise results, but took a huge time. With 18 galleries and 18 queries, it took around 20 minutes, and with 100 galleries and 1450 queries in the CCDA dataset, it took hundreds of hours and made this experiment impossible to perform.

LSP's original algorithm has a part where it obtains feature points and uses these feature points only for indexing and querying. Its intuition is only select points that have maximum or minimum shape index compared to its neighbor. This is a very good way as it will capture the most feature points, and remove points that are unimportant, for example, points on a flat surface. However, if we apply this method in each LSP separately like its original algorithm, we will get a different set of feature indices at different meshes. For example, imagine that at mesh A, we get feature points at indices 1, 2, and 3. However, at mesh B, we get feature points at indices 4, 5, and 6. This will make it very hard to match points between meshes because our target is matching points at the same index of different meshes.

To solve this problem but still keep the original idea of the LSP algorithm, we repeat the process of getting feature points on many meshes just to get the indices of feature points, and then we accumulate these feature points index among different meshes. By applying this method to a huge number of meshes, we get a total of 172 feature indices. Then we will use index and query points at these feature indices only instead of using all 10475 points. In the experiments chapter, we will show that by using only 1.6% number of points, we still get really similar results compared to using all 10475 points, while reducing

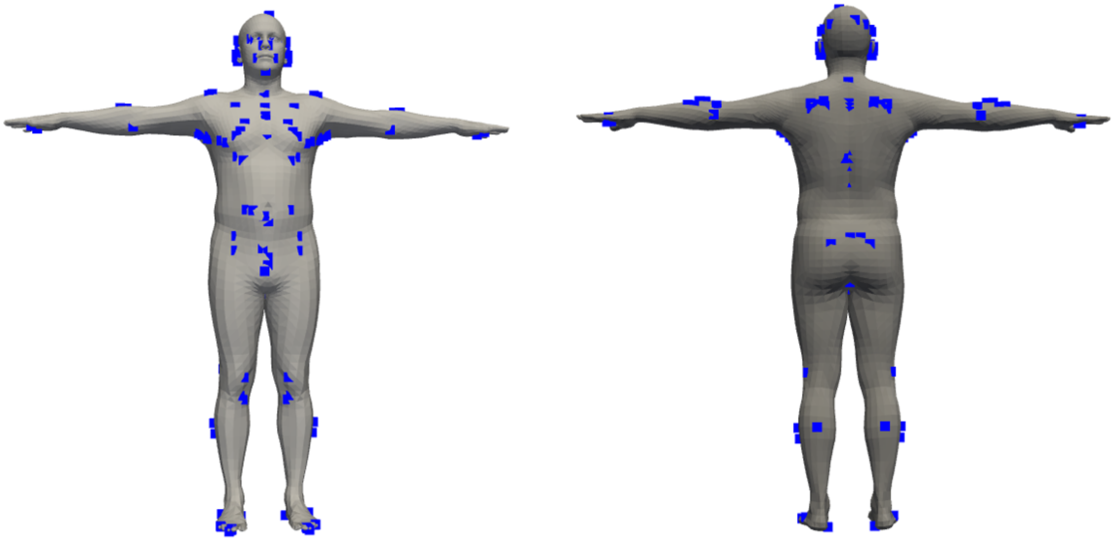


Figure 3.3: 172 feature points location on the 3D mesh

the huge time for indexing and querying and making our experiments with large datasets feasible. You can see locations of these 172 indices in 3D mesh in Figure 3.3

Algorithm 1 *Get feature indices*

```
$F \leftarrow$ empty set
for each mesh $M$ in list of meshes do
    for each vertex $P$ on $M$ do
        Feature indices $\leftarrow$ empty set
        $S_i \leftarrow$ shape index at $P$
        $N \leftarrow$ list of points in radius $r$ from $P$
        $S_{max} \leftarrow$ max shape index of all point in $N$
        $S_{min} \leftarrow$ min shape index of all point in $N$
```

	In range image (LSP paper)	**In 3D mesh (Our approach)**
Input	Images with pixel value represent the depth of an object.	3D vertices, edges, and faces.
Compute curvature	Fit quadratic surface, use differential geometry.	Angle deficit method, cotangent formula.
Viewpoint	Can get two different images of the same object with different viewpoints $->$ easy to verify.	Cannot get two different presentations of the same object with different viewpoints, because 3D mesh can rotate $->$ need to find a way to verify.

Table 3.1: Differences in Computing LSPs

```
        if $S_i == S_{max}$ or $S_i == S_{min}$ then
            Feature indices add index of P
        end if
    end for
    F add Feature indices
end for
```

3.4 Compute LSPs from A 3D Mesh

3.4.1 Differences in computing LSPs compared to LSP paper

Compute LSPs in 3D mesh are different compared to compute LSPs in range image in LSP paper as you can see in Table 3.1.

3.4.2 Some Changes to LSP Original Papers

Histogram vs KDE

In the LSP[4] paper, the 2D histogram is formed from shape indices and cosine angles of all neighbor points. However, the histogram is very sensitive to bin size and causes some incorrect matching in our experiments. Also, shape index and cosine of angle are continuous variables, so it does not really make sense to create a histogram. We replaced histogram with kernel density estimation (KDE) and got very stable and smooth results when matching LSPs.

Shape index

In the original LSP paper, $S_i \leftarrow \frac{1}{2} - \frac{1}{\pi}\tan^{-1}\left(\frac{k_{\max}+k_{\min}}{k_{\max}-k_{\min}}\right)$. Therefore, $Si \in [0,1]$. However, the cosine angle is $\in [-1,1]$. Due to this range mismatch, the LSP author uses 17 bins for Si, and 34 bins for cosine angle. We changed to $S_i \leftarrow \frac{2}{\pi}\tan^{-1}\left(\frac{k_{\max}+k_{\min}}{k_{\max}-k_{\min}}\right)$, so that $Si \in [-1,1]$ and then we can use the same number of bins for both S_i and cosine angle.

Checking surface type

In LSP paper, $sgn_{\varepsilon_x}(X) = \begin{cases} +1 & \text{if } X > \varepsilon_x \\ 0 & \text{if } |X| \le \varepsilon_x \\ -1 & \text{if } X < -\varepsilon_x \end{cases}$

Surface type $T_p \leftarrow 1 + 3\left(1 + \mathrm{sgn}_{\varepsilon_H}(H)\right) + \left(1 - \mathrm{sgn}_{\varepsilon_K}(K)\right)$

When we matched LSP and visualized the failed cases, we saw that many cases of LSPs cannot match because they have small differences in H or K but then the sign of H or K is different, leading to them having different surface types. For example, two points have the same local shape, point 1 has $H = -0.001, K = 165$ and point 2 has $H = 0.001, K = 165$. As you can see, they have small differences in both H and K. If we set $\varepsilon_H = \varepsilon_K = 0$, we will get different $\mathrm{sgn}_{\varepsilon_H}$, which lead to different T_p for point 1 and point 2. However, even if we change ε_H and ε_K to a different number, we will get the same problem. For example, we set $\varepsilon_H = \varepsilon_K = 0.003$, we got a case that point 1 has $H = -0.0029, K = 165$, and point 2 has $H = 0.0031, K = 165$ that lead to different T_p. Therefore, we removed surface type checking and saw that our LSP matching results are improved.

Index and query

LSP paper [4] described a method to index and query. This method computes mean (μ) and standard deviation (σ) for each shape index in neighbor points N. Then a hash table is created, with each location in this hash table is a bin for μ and σ. For each LSP, μ and σ are retrieved to place this LSP into the bin where μ and σ fall into.

This way of hashing seems ideal for indexing and fast retrieval, however, it has many problems:

- First, it's based on a mean and standard deviation of the shape index of all points in the neighbor region. If 2 LSPs have the same histogram, they will have the same μ and σ deviation and fall into the same bin. However, it's not strictly in the reverse way. Two LSPs that have the same μ and σ don't ensure they will have the same LSP. This means one bin can have multiple LSPs but their histograms are not close to each other, which makes this index method inefficient.

- Secondly, there will be a case when two histograms are really close to each other, also the same for their μ and σ, but due to a small number of differences, they can fall into different bins that are next to each other. This will lead to cases where we cannot match a histogram to those that have their distance smaller.

- Thirdly, we also visualize the number of each bin in this hash table when we try this method. Because the shape index is in the range [0, 1], so its mean is in the range [0, 1], and its standard deviation is in the range [0, 0.5]. We divided the mean and standard deviation into 100 bins each, then we created this hash table, looped into each object in the gallery, got the LSP of each point in each object, and put these LSP into bins of that hash table. However, as you can see in Figure 3.4, the x-axis is 100x100 bins, and the y-axis is the number of points falling in each bin, the number of points falling in each bin is not equal and highly different between bins. This makes this hash table inefficient for index and retrieval.

- Finally, because our final purpose is to compare the difference in the LSP histogram between points in a query object and points in a gallery object, we can flatten this histogram, and then index and query this using vector nearest search, a popular way to index and query vector that many research for recognition used. We will use this way, combined with checking surface type in the LSP paper as our initial method for index and retrieval.

We summarized all of our changes in Table 3.2.

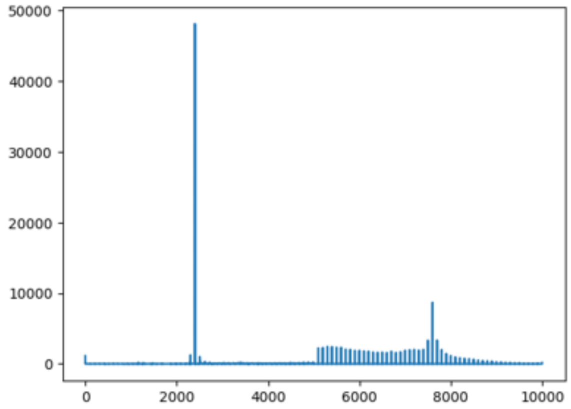


Figure 3.4: Number of points falling in each bins

LSP paper	**Our method**
2D histogram	2D kernel density estimation
Shape index $\in [0, 1]$	Shape index $\in [-1, 1]$
Check surface type	Remove surface type checking
Get feature points dynamically	Get feature points indices statically
Index and query using a hash table based on mean and std of shape index	Index and query using flatten KDE and vector nearest search

Table 3.2: Differences in Computing LSPs

3.4.3 Our LSP Implementation

Algorithm 2 *Our LSP Implementation*

```
for each vertex P in object do
    $k_{max}, k_{min} \leftarrow$ Principle curvature
    $S_i \leftarrow \frac{2}{\pi} \tan^{-1}\left(\frac{k_{max}+k_{min}}{k_{max}-k_{min}}\right)$
end for
for each point P in Feature points do
    $n_i \leftarrow$ vector normal at point i
    $N \leftarrow$ list of points N satisfy: $\|N - P\| \leq \varepsilon_1$
    $KDE \leftarrow (S_i, \cos(n_p \cdot n_n))$ for all points in N, $S_i \in [-1, 1]$, $\cos(\mathbf{n}_p \cdot \mathbf{n}_n) \in [-1, 1]$
    Centroid $C \leftarrow$ centroid of all points in N
    P embedding $\leftarrow$ L2_norm(flatten(KDE))
end for
```

Chapter 4

Experiments and Results

In this chapter, we discuss our implementation details and parameters in Section 4.1 and our evaluation metrics in Section 4.2. Then we discuss how we verify our LSP implementation in Section 4.3. Then we show detailed results of our LSP method on a small dataset by using 18 galleries and 18 queries of 18 people in the HBW dataset in Section 4.4. Next, in Section 4.5, we tried our LSP method with a large dataset. This dataset includes 100 galleries and 1450 queries of 100 people in the CCDA dataset. In the next Section 4.6, we visualize the results of our experiment and explain these results. We also perform our LSP method on 100 galleries and 100 queries with the same images as galleries but head occluded in Section 4.7 to see how our method performs in case the body in the images is occluded. In Section 4.8, we discuss some possible reasons for our results in previous sections. In Section 4.9, instead of using only one image of each subject in the gallery and query set, we use multiple images and average their SHAPY shape parameters. In section 4.10 and 4.11, we use data augmentation combined with average shape parameters. In section 4.12 we summarize the results of our experiments, and section 4.13 will visualize and quantify the effect of averaging shape parameters. Finally, we show our SHAPY virtual measurements results on self-images in Section 4.14.

4.1 Implementation and parameters

4.1.1 Implementation

To obtain 3D mesh from SHAPY, we need an RGB image and the pose of the person in that image. We use Openpose to get this pose results.

All of our implementations are written in Python. We used the libigl library for reading the mesh vertices and faces, and also for getting direct principal curvature k_{max} and k_{min}. This reduces the risk of getting H (mean curvature) and K (Gaussian curvature) first, because we already tried a lot of libraries including trimesh, libigl, or pymesh for getting H and K, and then we cannot compute k_max and k_min at a lot of vertices because $H^2 - K < 0$. libigl makes it easy for us to get directly k_{max} and k_{min} using fucntion `igl.principal_curvature`. Then from k_{max} and k_{min}, we compute shape index S_i at each vertices. We used `np.arctan2` for compute $\tan^{-1}\left(\frac{k_{\max}+k_{\min}}{k_{\max}-k_{\min}}\right)$. We used `KDTree.query_ball_point` to get neighbor indices of each vertex in the mesh, and used function `igl.per_vertex_normals` for getting normal at each vertex. Then we obtain angles by computing the dot product between vertex normal at the feature point and vertex normal at all points in neighbor indices. We used `sklearn.neighbors.KernelDensity` to get a KDE from the shape index and angles of all points in the neighbor region. Then we flatten our KDE, normalize using L2 norm, and use `faiss` library to index and query vectors.

We got feature indices by using the algorithm for getting feature points in LSP paper, with $\alpha = \beta = 0$ because we don't want to filter more points and fine-tune these hyper-parameters. Then we run through a lot of meshes and accumulate these indices to get the final 172 feature indices. We also print out the number of feature indices each time it loops through a mesh, and we see that at the end of this process, it seems to converge at 172 points.

4.1.2 Parameters

Because we removed surface type checking, and also we removed a lot of parameters from the LSP original algorithm, currently we only have $\epsilon_1 = 0.1$, which is the radius of the neighbor region, and the numbers of bins in both axes are 10, so feature vector for each LSP has length 100. We have tried several different values of ϵ_1 and the number of bins, but those values seem mostly suitable for SHAPY mesh and our experiment.

4.2 Evaluation metrics

For evaluation metrics, we use top-1 accuracy and mean average precision (mAP). We also plot the cumulative match curve (CMC) for our results. When we evaluate our method on a dataset, we evaluate it at both point level and subject level, because our LSP algorithm indexes and queries each feature point in both gallery subject and query subject. Point level means with each point index i^{th} in a query subject of person P, point index i^{th} at the gallery subject of person P is considered a correct hit. We got the subject level result by aggregating the top-1 results of each feature point in each query subject.

4.2.1 Top-1 accuracy

Top-1 accuracy is a metric that is mostly used in recognition and identification systems. It evaluates how many percent of queries the recognition system obtains the ranking whose the top 1 is the correct result. For our experiments, with each query subject, a query is considered as a feature vector (combined vector of histogram and FC vector) for each feature point. Then from this query vector, we query the nearest vector in the database we indexed before using these feature vectors for all gallery subjects. Then we get each ranking for each query, with the top-1 in this ranking being the nearest vector compared to the query vector, top-2 is the second nearest, and so on. If a query vector from a feature point index i^{th} in a query subject of person P, a correct hit (the nearest vector or top-1 result) should be the feature vector at feature point index i^{th} in a gallery subject of person P.

After we get all ranking results for all feature points in a query subject, we aggregate the top-1 result of these rankings for the subject level results. For example, there are 172 feature points in a query subject Q_1. After getting the ranking of each feature point, we get 140 top-1 results are vectors of 140 points at gallery subject G_1, 20 top-1 results are vectors of 20 points at gallery subject G_2, 10 top-1 results are vectors of 10 points at gallery subject G_3, 2 top-1 results are vectors of 2 points at gallery subject G_4. Then at subject level, with query Q_1, we got ranking G_1, G_2, G_3, G_4 (sorted by 140, 20, 10, 2). If gallery subject G_1 is the same person as query subject Q_1, this be considered a top-1 hit for this query. We repeated this to get the result of all query subjects and got the top-1 accuracy for the subject level result.

4.2.2 mAP

Mean average precision (mAP) is a more comprehensive metric to evaluate the retrieval system and is also mostly used in recognition and identification systems too. For each ranking, it compute the precision at the rank of the correct hit, and then average this to get the average precision for each query.

For example, with point index i^{th} of query subject person P, we got ranking $R_1, R_2, ..., R_n$. At rank R_3, R_5, R_{11} we got a correct hit which is a vector at points index i^{th} of 3 gallery subjects of person P (assume we have 3 gallery subjects, all of them are person P). The precision at rank 3^{th} is $1/3$, which is the number of correct hits at that rank divided by the number of predictions. We get the sum of precision at rank 3^{th}, 5^{th}, 11^{th}, which is $1/3 + 2/5 + 3/11$, and then we divide by 3 to obtain the average precision for this query. With each query, we repeat this process to get average precision, and then we get the mean of all average precision for all queries, which is the mean average precision (mAP).

This metric is very useful for cases where with each query, there all multiple cases in the gallery that can be considered for correct hit. If we can push all of these correct hits to the top of the ranking for all queries, we can get the maximum mAP.

Unfortunately for most of our experiments, we used only 1 3D mesh of 1 person in the gallery, so it's a little bit cannot extract more meaning from this metric. However, because this metric is popular in recognition, we still use it in our experiments.

4.2.3 CMC

The cumulative match curve (CMC) is a curve that shows all of the top-n accuracy. For example with our CMC plot, we show all of top-1 accuracy to top-50 accuracy. This metric is very useful to compare different approaches, where approach A_1 can get higher top-1 accuracy than approach A_2 , but later then, from top-10 accuracy to top 50 accuracy, approach A_2 get higher accuracy than approach A_1.

4.3 Validate LSP implementation

Validate LSP implementation is a very important step in our research. It is the foundation of all experiment results, because if it's incorrect, then none of the results correct.

4.3.1 Implementation differences and difficulties when verification

There are some differences when getting LSP in range image (original LSP paper) and 3D mesh (our implementation):

- The first difference is in data representation. A range image is represented by a 2D grid of values representing the depth of an object. On the other hand, a 3D mesh is represented by vertices and faces.

- Because of the difference in data representation, compute curvature is totally different between range image and 3D mesh. In the LSP paper, they fit a quadratic surface $f(x, y) = ax^2 + by^2 + cxy + dx + ey + f$ to a window centered at each point in the range image. This quadratic surface fitting method has high complexity. In many 3D libraries, they use the angle deficit method for getting Gaussian curvature, and the cotangent formula for getting mean curvatures.

- We can verify LSP in range images by using two images of the same subject but different viewpoints (like the way the author of LSP paper did) to verify. However, in 3D mesh, we cannot obtain two presentations of two viewpoints on the same subject. A 3D mesh can be rotated to any angle and we can see it from many viewpoints.

It took us a long time to think about how we could verify our LSP implementation. We discuss two methods in two subsections below, which are based on the characteristic of LSP that it captures the local shape feature from feature points, so if LSP has the same histogram, they should have the same local shape and vice versa.

4.3.2 Using a manually created 3D subject to verify

The first method we used to verify LSP was using a manually created 3D subject. We created a tetrahedron, with 4 vertices O(0, 0, 0), A(1, 0, 0), B(0, 1, 0), C(0, 0, 1). For each vertex, we obtained its LSP using it as the feature point and the three remaining vertices as neighbor points. By definition and the characteristic of LSP which is to capture the local shape feature, it is obvious that the histogram at A, B, and C should be equal, and different from the histogram at O. As you can see in Figure 4.1, our LSP implementation indeed reflects this, which is proof that our LSP implementation is correct. We also got H and K of A, B, and C equally while H and K from O are different; and got angles of a vector normal between O and each vertex A, B, C equally, and different with the angle of vector normals between A and B, B and C or C and A.

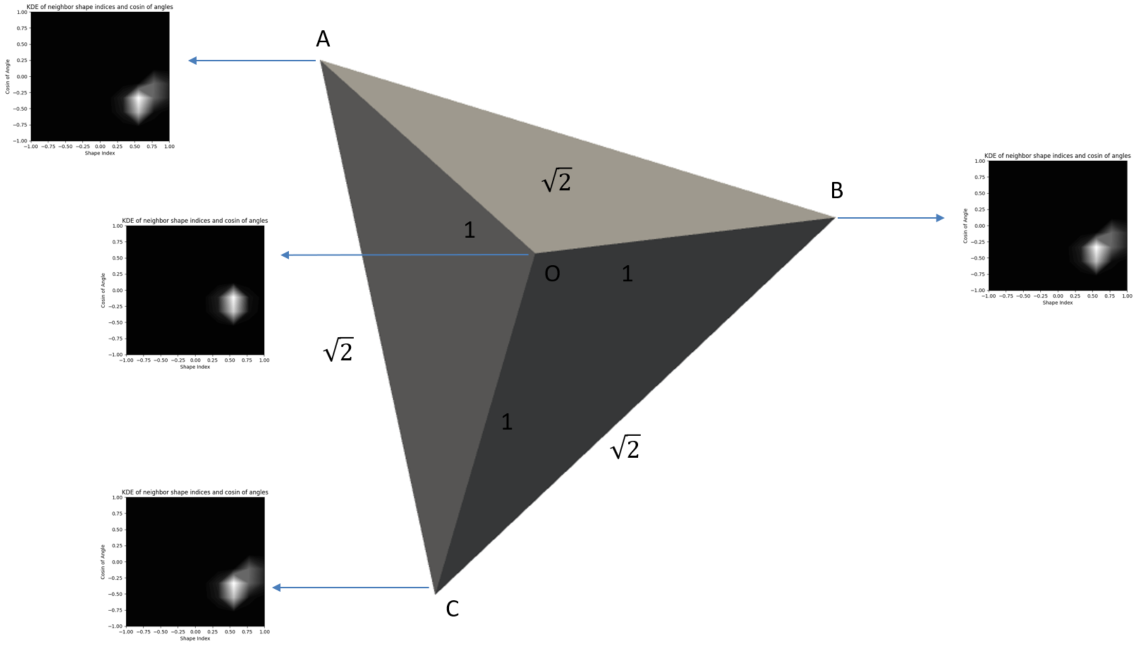


Figure 4.1: Verify LSP implementation using a tetrahedron

4.3.3 Verify by matching LSPs at two mesh of the same person

Testing LSP using manually created 3D subjects seems not comprehensive enough, so we also test LSP implementation by matching LSPs at two mesh M1 and M2 of the same person but in different images, as we discussed in the iterative process in chapter 3. When querying a point index i at mesh M_1, a correct hit is a point index i at mesh M_2. We got 63% top-1 accuracy and 100% top-2 accuracy when matching LSPs in this case, proving that LSP implementation is working correctly and that each LSP reflects the local shape of each position in the human body. (failed cases in top 1 are symmetric on the 3D body, therefore, at top 2 we have 100% accuracy).

4.4 Experiment 1: 18 people

4.4.1 Dataset obtain and pre-processing

Human Bodies in the Wild (HBW) is a dataset that is presented in the SHAPY paper. It has RGB pictures of ”photo-lab” settings and in-the-wild photos of 35 subjects, with 10 subjects in the validation set and 25 subjects in the test set. In 25 subjects in the test set, we got 18 subjects who got a ”photo-lab” setting and got 2 photos of each, one is for our gallery and one is for our probe (query). Because of all them are photo-lab settings, so all of them have no background, simple camera view, and simple pose, and two images with the same person always have the same clothes. We want to eliminate as much as factors that can cause noise for regressing the 3D human body from images using SHAPY.

Then we get the openpose output of these 36 images, get the SHAPY canonical pose, get the LSP of each feature vertices (among 172 feature indices), and get feature vectors (combined vector) of each LSP. We index all of the feature vectors obtained from 18 images in the gallery and then use feature vectors obtained from 18 images in the probe as our query.

4.4.2 Results

Our results at both point level and subject level are in Table 4.1. Also, we obtained a CMC plot for point level result in Figure 4.2 and subject level result in Figure 4.3.

We can see several things from those results:

	Point level		subject level	
	Rank1	mAP	Rank1	mAP
172 feature points	**14.99%**	**30.53%**	33.33%	**49.46%**
All 10475 points	14.56%	28.80%	33.33%	48.40%

Table 4.1: Results on 18 galleries and 18 queries from 36 ”photo-lab” setting images of HBW dataset

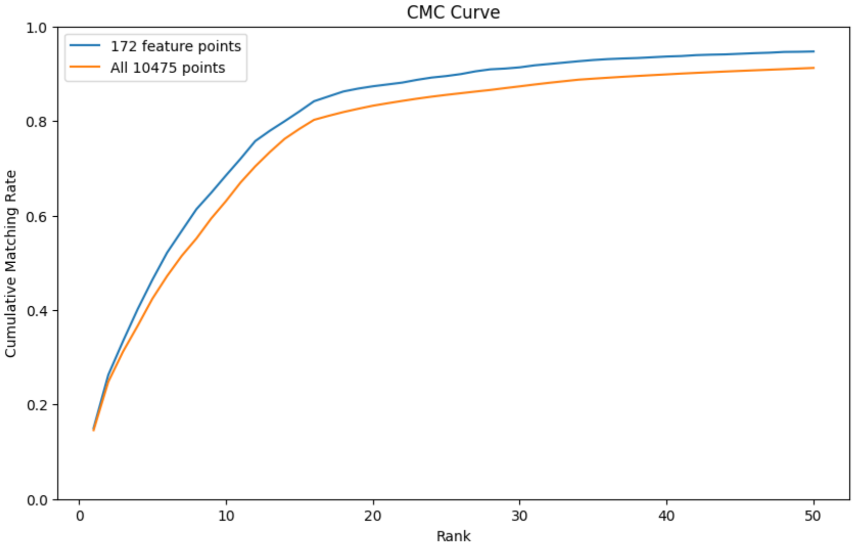


Figure 4.2: CMC plot for point level results on 18 galleries and 18 queries from HBW dataset

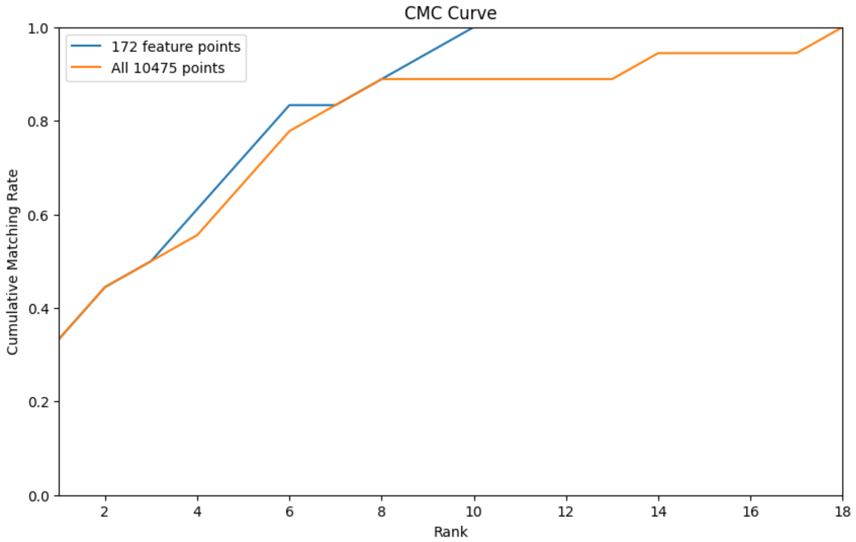


Figure 4.3: CMC plot for subject level results on 18 galleries and 18 queries from HBW dataset

- Using 172 feature indices achieves similar, even slightly better results compared to using all points. This means 172 feature indices indeed capture the most feature points on subjects while using only 1.6% number of points.

- Using 172 feature indices got better top accuracy than using all points at both point level results and subject level results at any rank as you can see in Figure 4.2 and Figure 4.3. This means using a smaller number of points seems to reduce the noise, kind of regularization effect, and get higher results.

- Even though this dataset is small, easy, and contains only photo-lab settings, our results are not good. We need to visualize and understand how we got results like this in Section 4.6.

4.5 Experiment 2: 100 people

4.5.1 Dataset obtain and pre-processing

CCDA (Clothing Changes and Diverse Activities) is a dataset from paper 3DInvarReID. This dataset contains images with diverse human activities and clothing changes for evaluating long-term re-identification. Specifically, this dataset contains popular athletes and popular artists, and with each subject, it has two sets of images: 'challenging' and 'normal' body poses. All images are bounding boxes of body region only and have been resized to 256 x 128. The gallery set, it has 100 images at normal poses, and in the query set, it has 1455 images at both normal and challenging poses.

We got open pose results of all images in the gallery and query, then we ran SHAPY and got 3D body meshes in a canonical pose from all images. Due to the open pose cannot detect pose in 5 images in the query set, we got a total of 100 meshes in the gallery and 1450 meshes in the query.

4.5.2 Results

For the CCDA dataset, running with all points is impossible because it took a huge time, so we only experimented using 172 feature indices. Our results are shown in Table 4.2, CMC plot for point level results at Figure 4.4 and subject level results at Figure 4.5. As you can see, these results are not good, so we need to see why we got these results and have an explanation for this in the next Section 4.6.

Point level		Subject level	
Rank1	mAP	Rank1	mAP
1.22%	5.15%	1.45%	7.54%

Table 4.2: Results on 100 galleries and 1450 queries of 100 people in CCDA dataset

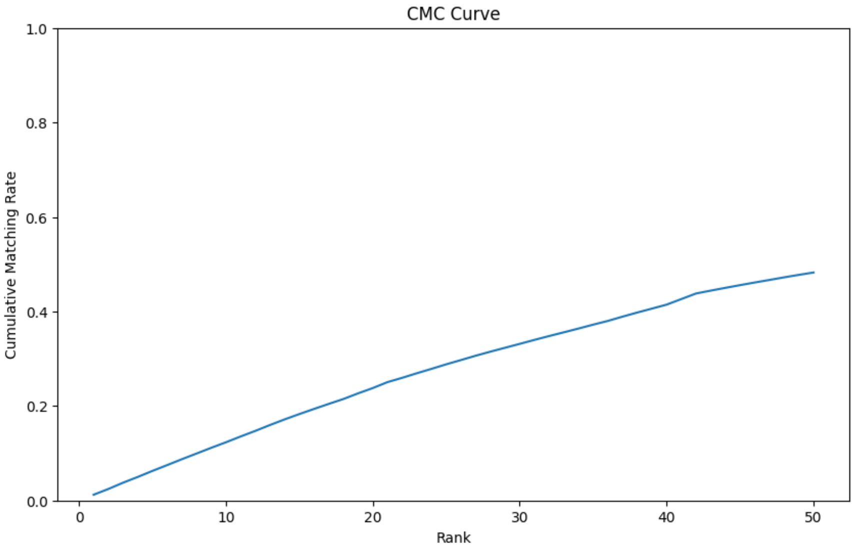


Figure 4.4: CMC plot for point level results on CCDA dataset

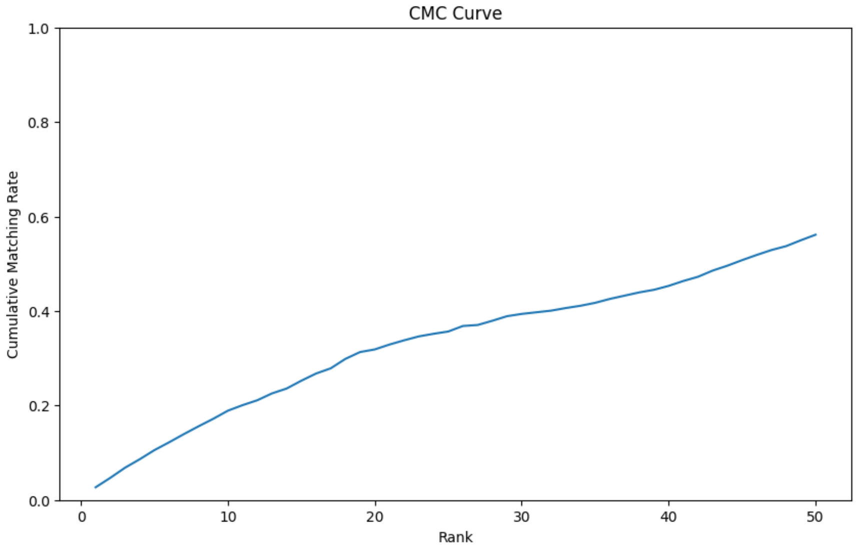


Figure 4.5: CMC plot for subject level results on CCDA dataset

4.6 Visualize and explain results

It's very hard to try to explain at first because most of the failed matching cases (point level) are the same index of 3D mesh. For example, in a test with 18 subjects of the HBW dataset, point index i^{th} at mesh 30t (mesh with subject id 30 in query) should be matched with point index i^{th} at mesh 30 (mesh with subject id 30 in gallery), but it matches with point index i^{th} at mesh 34 (mesh with subject id 34 in gallery). Because they are all at the same index, when visualizing their shape, it's very hard to see the difference with our eyes. Even when we check with a histogram of their LSPs, they have really similar histograms and we cannot differentiate visually, we only compute the L2 distance and got the distance of histograms between mesh 30t and mesh 34 is smaller than the distance of histograms between mesh 30t and mesh 30. We need a way to visualize them and can explain the results.

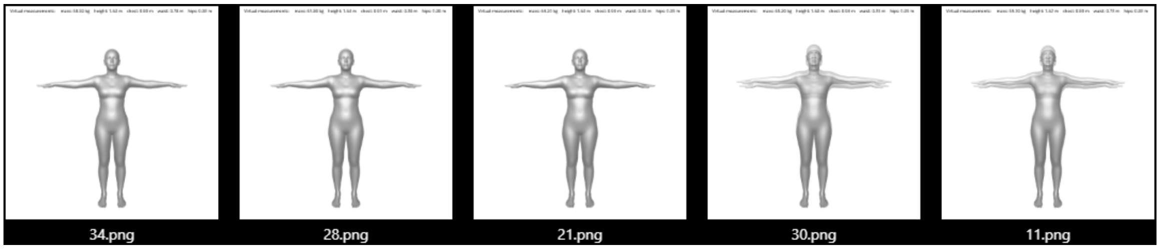


Figure 4.6: Top 5 results (from left to right) when we query mesh 30t, each image is fused with image 30t and the image at the ranking itself.

4.6.1 Visualize using images of mesh in canonical pose

When we get SHAPY 3D mesh in the canonical pose, we also can get an image of this canonical pose. Figure 4.6 showed the top 5 results (from left to right) when we query mesh 30t, each image is fused with image 30t and the image at the ranking itself. As you can see, the top 1 result is not blurry, it means the body of mesh 30t matches with the body of mesh 34 visually. When we move to a higher ranking, especially rank 4^{th} and 5^{th}, we can easily see the blur in the body, it means the body of mesh 30 is highly different compared with the body of image 30 (rank 4^{th}) and image 11 (rank 5^{th}) in term of visual.

With this visualization, we can clearly see that:

- LSP captured the human body shape and reflect this in the LSP matching results.
- We can see that the SHAPY results from subject 30 are inconsistent. Even though the mesh from images 30t and 30 is the same person, SHAPY outputs their 3D mesh body that is highly different visually, so that is the reason why the 3D body of image 30t is matched with the 3D body of different images. This is not only for this query (mesh 30t), but also for all another query. That is why we have only 33.33% top 1 accuracy with 18 subjects in HBW and 2.69% top 1 accuracy with CCDA.

Mesh	Mass(kg)	Height(m)	Chest(m)	Waist(m)	Hips(m)
30t (query)	59.30	1.62	0.90	0.78	0.98
34 (rank 1)	58.52	1.63	0.89	0.78	0.97
28 (rank 2)	61.88	1.64	0.91	0.80	1.00
21 (rank 3)	64.21	1.64	0.94	0.82	1.01
30 (rank 4)	65.20	1.68	0.94	0.81	1.01
11 (rank 5)	59.10	1.67	0.89	0.77	0.97

Table 4.3: Virtual measurements of mesh query image 30t and meshes of its top 5 ranking.

4.6.2 Using virtual measurements from SHAPY meshes to explain failed cases

In Section 2.3.4, we mentioned that SHAPY built a model that predicts virtual measurements from a 3D mesh. They also used this model to train their model based on the virtual measurement ground truth of internet model images. Then we use this model to get virtual measurements from SHAPY 3D mesh to see the differences in 3D body shape. Table 4.3 showed virtual measurements of mesh query image 30t and meshes of its top 5 ranking. We can clearly see that mesh 34 has measurements really similar to mesh 30t (that is why it's top 1), while later mesh at later rank has measurements highly different compared to measurements of mesh 30t. This also confirmed that SHAPY results from subject 30 are inconsistent not only in visual but also in virtual measurements.

4.6.3 Using betas (shape) parameters from SHAPY meshes to show that LSP strongly capture shape feature of objects

SHAPY canonical pose mesh is obtained from shape (betas) parameters only. We want to see is there any relationship between SHAPY shape parameters and our LSP ranking.

Rank1	Rank2	Rank3	Rank4	Rank5	Rank6	Rank7	Rank8
1.746998	2.025983	2.204594	2.349410	2.455906	2.631209	2.674781	2.732079

Table 4.4: L1 distance average of query shape parameters and shape parameters of each objects in ranking.

Let's call β_i the shape parameter of the mesh at query i^{th}, β_{ij} the shape parameter of the mesh at rank j^{th}. I compute $S_j = \frac{1}{Q}\sum_{i=1}^{Q}|\beta_i - \beta_{ij}|$ with Q is the number of queries. Table 4.4 shows S_1 to S_9 for LSP results from the CCDA dataset. We can clearly see that our LSP ranking clearly has a strong relationship with SHAPY shape parameters: the L1 distance between query shape parameters and shape parameters of objects at high rank like rank1 or rank2 is smallest, and then the rank higher, the L1 distance higher. This is very strong proof that our LSP methods are able to capture and differentiate the different shapes of objects.

4.7 Experiment 3: 18 people, query same images as gallery but head occlude

As you know, from RGB images, SHAPY use a deep learning model to get the shape and pose parameters, and then from these parameters, they use the SMPL-X model to generate 3D mesh. So even if a head is occluded from an image, SHAPY still can produce full 3D human body mesh. This characteristic is ideal for recognizing humans in images where some part of the human body is missing. We want to see if we occluded heads from people in images, what results we can get using LSP?

Point level		Subject level	
Rank1	mAP	Rank1	mAP
19.06%	35.07%	38.89%	55.89%

Table 4.5: Results on 18 galleries and 18 queries with the same images as galleries but head occluded

4.7.1 Data pre-processing

We still used 18 images in the HBW as our gallery, however with the query set, we still used the same images in the gallery (to reduce the effect of predicting different shape parameters) but we occlude the head of the person in each image. We used `cv2.CascadeClassifier.detectMultiScale` to detect the head region of the person in each image, then we fill detected face regions with white color (because our image's background is white). Then we got a gallery with 18 images, and query with 18 images in the gallery but head occluded.

4.7.2 Results

Our rank1 and mAP results are at Table 4.5, CMC plot for point level result at Figure 4.7 and CMC plot for subject level result at Figure 4.8. As you can see, these results are not high. We need to check why we have these results.

As you can see in Figure 4.9, even if we only occlude the head region, SHAPY can produce high differences in shape parameters and lead to high differences in virtual measurements. This again confirms that shape parameters produced from SHAPY are highly variance with 2 images of the same person.

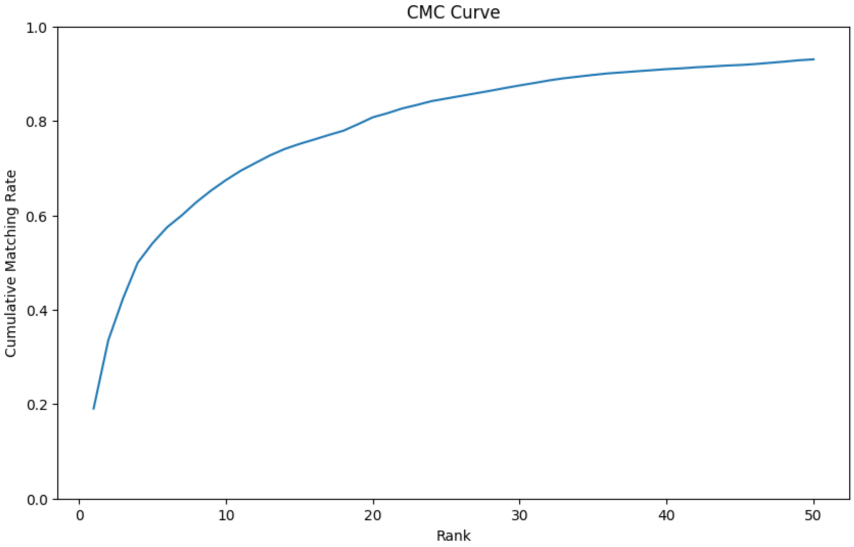


Figure 4.7: CMC plot for point level results on with 18 head-occluded queries

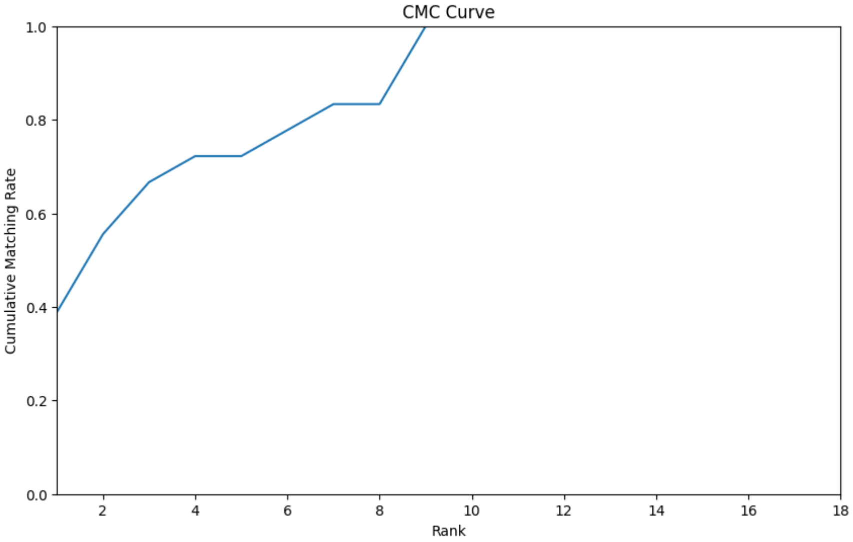
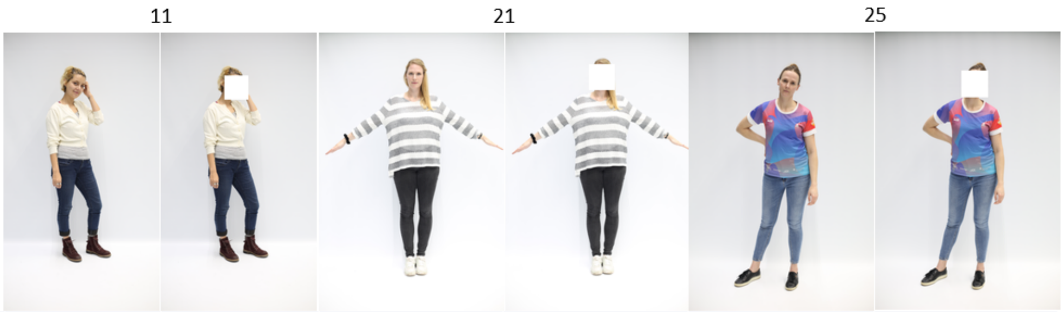


Figure 4.8: CMC plot for subject level results on with 18 head-occluded queries



Mass	59.10kg	71.68kg	64.21kg	73.63kg	56.49kg	61.38kg
Height	1.67m	1.77m	1.64m	1.75m	1.61m	1.65m
Chest	0.89m	0.98m	0.94m	1.01m	0.88m	0.92m
Waist	0.77m	0.83m	0.82m	0.86m	0.77m	0.81m
Hips	0.97m	0.98m	1.01m	0.98m	0.96m	0.96m

Figure 4.9: Virtual measurements with head occluded

4.8 Possible reasons why SHAPY produces different shape parameters for images within the same person

SHAPY consists of two components: deep learning model to get shape and pose parameters, and SMPL-X model to get 3D body mesh from these parameters.

In our test with 18 subjects from the HBW dataset, all the images we used have no background, the same clothes, and the same camera view for each pair of images with the same person, so at the beginning, we might think that the reason why SHAPY produce high variance shape parameters is because of the pose of the person in an image. However, in the occluded test, we can see that even though two images are the same, with the pose the same too, and only the head occluded in one image, SHAPY can produce highly different shape parameters. This outcome showed that SHAPY seem biased, or overfit on the dataset they used to train, and they didn't use data augmentation to make the model robust for cases like head occluded.

To improve it, we might need to re-train this deep learning model to make it predict shape much more consistently for the same person. We can use some loss to keep the shape parameters the same for multiple images of the same person [29], or we can use data augmentation to occlude random parts of the human body and make the model still produce consistent shape parameters in these cases.

4.9 Experiment 4: 18 people, averaging shape parameters using multiple different images

As we can see due to some random noise in the image, SHAPY can produce incorrect shape parameters. What if we use multiple images of the same person in both gallery and query set, and then average SHAPY shape parameters from their meshes to reduce the errors that SHAPY causes?

4.9.1 Data pre-processing

This time, instead of choosing two images (one for the gallery, one for the query) for each subject in the HBW test set, we use 5 images for the gallery and 5 images for the query. Then for images of the same subject in both the gallery and query set, after running SHAPY and getting shape parameters, we average these shape parameters, get 3D mesh from those shape parameters, and use this mesh as a representation of that subject in the gallery or query.

	Point level		Subject level	
	Rank1	mAP	Rank1	mAP
One image per subject	14.99%	30.53%	33.33%	49.46%
Average shape parameters with multi images	**22.00%**	**39.78%**	**44.44%**	**59.54%**

Table 4.6: Results of experiment 4 compared with results of experiment 1

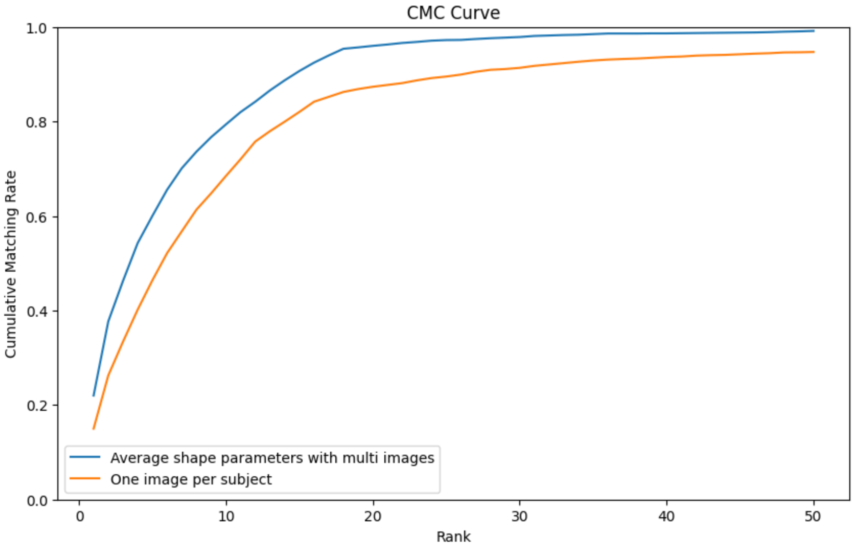


Figure 4.10: CMC plot for point level results comparing one image per subject method and average shape parameters method

4.9.2 Results

Our top 1 accuracy, mAP results, and CMC plot for both point level and subject level are in Table 4.6, Figure 4.10, and Figure 4.11. We also added in the table and figures the results of one image per subject method we got in Section 4.4. As you can see, we got much higher results using this average shape parameters method. This shows the effectiveness of reducing errors of shape parameters from SHAPY using multiple images in both gallery and query set.

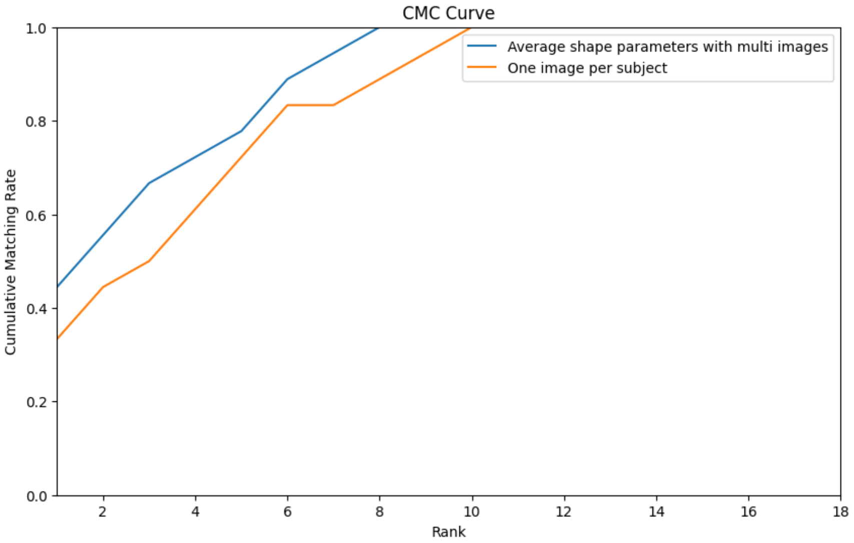


Figure 4.11: CMC plot for subject level results comparing one image per subject method and average shape parameters method

4.10 Experiment 5: 18 people, averaging shape parameters using multiple head-occluded images in query set

We also want to see the performance of averaging shape parameters using multiple head-occluded images in the query set.

4.10.1 Data pre-processing

This time, we use the same gallery as experiment 4. For each image in the query set of experiment 4, we use `cv2.CascadeClassifier` to get the head region of the person in the image and fill detected head regions with white color. Then for images of the same subject in both the gallery and query set, after running SHAPY and getting shape parameters, we average these shape parameters, get 3D mesh from those shape parameters, and use this mesh as a representation of that subject in the gallery or query.

Point level		Subject level	
Rank1	mAP	Rank1	mAP
12.63%	27.62%	16.67%	40.80%

Table 4.7: Results of experiment 5

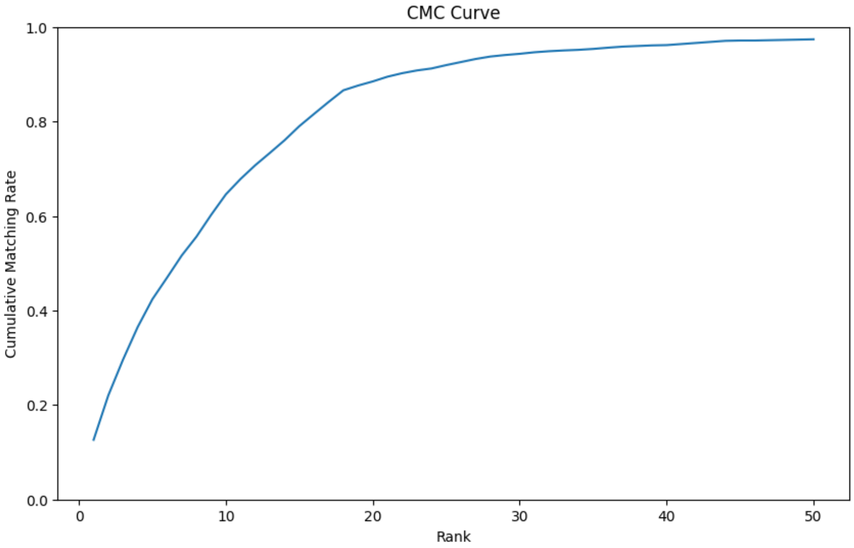


Figure 4.12: CMC plot for point level results of experiment 5

4.10.2 Results

Our top 1 accuracy, mAP results, and CMC plot for both point level and subject level are in Table 4.7, Figure 4.12, and Figure 4.13. Seems when the head occluded, the variance is too high and added up.

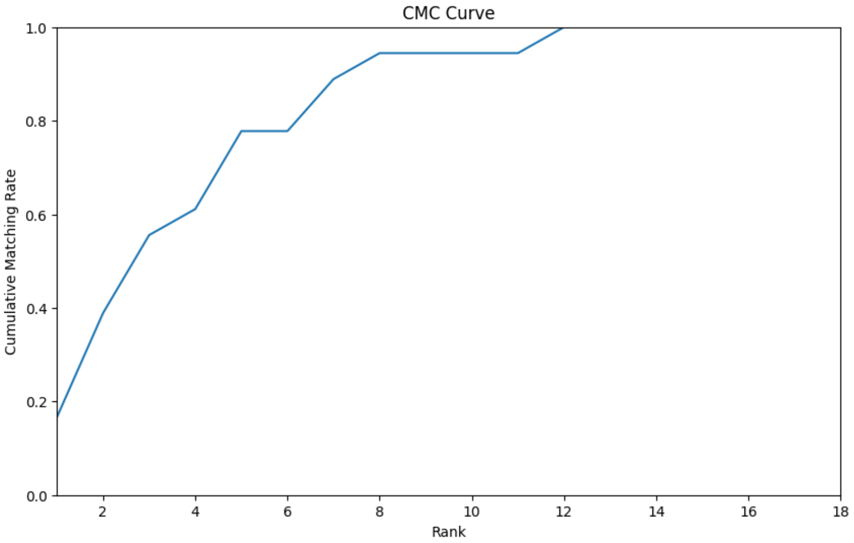


Figure 4.13: CMC plot for subject level results of experiment 5

4.11 Experiment 6: 18 people, averaging shape parameters using multiple parts occluded images

If we get multiple different images of each person in the gallery and query, then we can do experiment 4. However, we do not always have multiple different images of each person. In this experiment, we want to use data augmentation on the query set, combined with averaging shape parameters like before to see whether it can have good results.

4.11.1 Data pre-processing

We use the same gallery as experiment 4. For each image in the query set of experiment 4, we use YOLOv9 [31] to detect the bounding box of a person and create 5 occluded images like Figure 4.14. We have 5 different images of each person, and with each

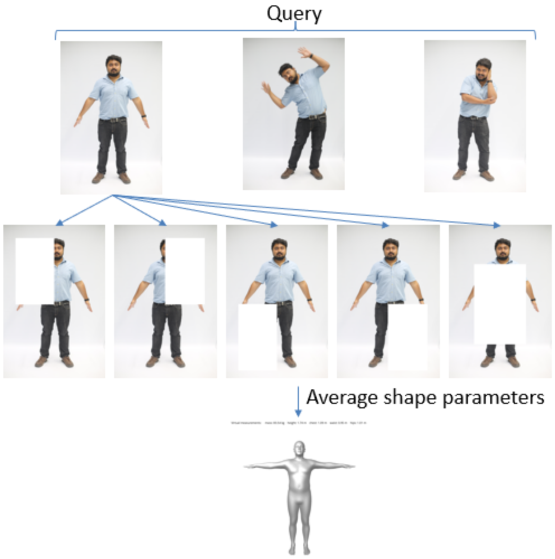


Figure 4.14: We use 5-part augmentation on each image of the query set, and average shape parameters of all images for one person

Point level		Subject level	
Rank1	mAP	Rank1	mAP
12.24%	29.93%	22.22%	45.65%

Table 4.8: Results of experiment 6

image, we get 5 occluded images, so in total we have 25 images per person in the query set.

4.11.2 Results

Our top 1 accuracy, mAP results, and CMC plot for both point level and subject level are in Table 4.8, Figure 4.15, and Figure 4.16. We also added in the table and figures the results of one image per subject method we got in Section 4.4. As you can see, we got much higher results using this average shape parameters method.

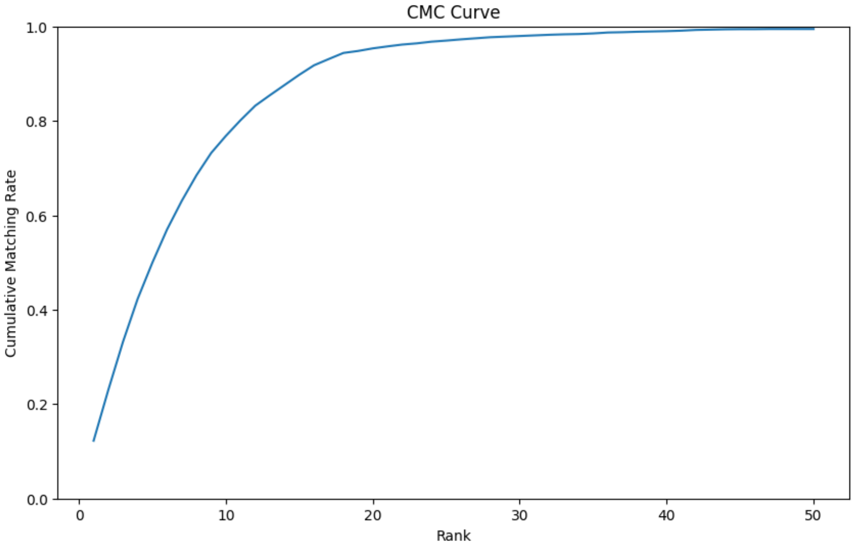


Figure 4.15: CMC plot for point level results of experiment 6

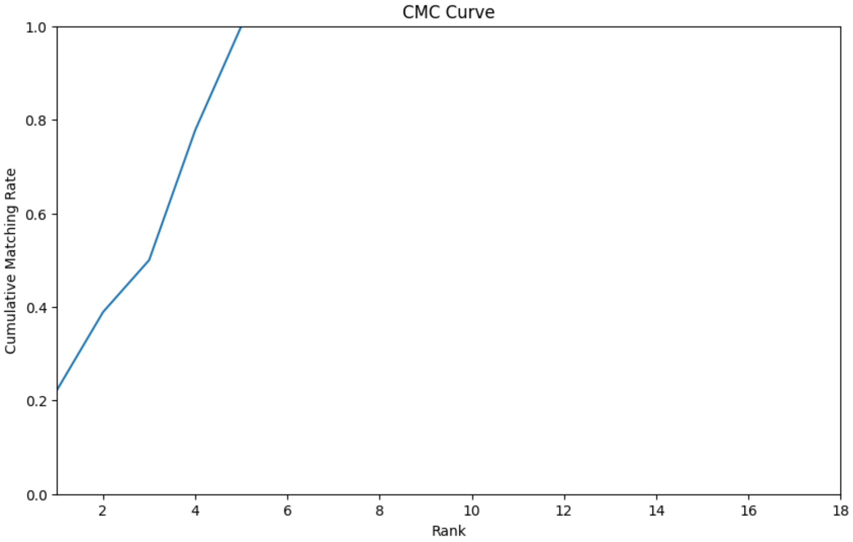


Figure 4.16: CMC plot for subject level results of experiment 6

Experiment	# ims in gallery	# ims in probe	# of subjects	Dataset	Rank1-sl
1 (sg)	18	18	18	HBW	33%
2 (CCDA)	100	1450	100	CCDA	2.68%
3 (sg-oc-head)	18	18	18	HBW	38.88%
4 (mt)	18 x 5 ims/sub	18 x 5 ims/sub	18	HBW	**44.44**%
5 (mt-oc-head)	18 x 5 ims/sub	same exp4	18	HBW	16.67%
6 (mt-oc-parts)	18 x 5 ims/sub	exp4 x 5 parts	18	HBW	22.22%

Table 4.9: Experiments summary (ims: images; sg: single; mt: multi; oc: occlude; sub: subject; exp: experiment; sl: subject level)

4.12 Experiment summary

Our experiment summary is in Table 4.9. Averaging shape parameters on multiple different images has the highest result. This shows the effectiveness of reducing errors of shape parameters from SHAPY using multiple images. We also used augmentation combined with averaging shape parameters but it didn't have high results. Seem using augmentation (at least two kinds we use) wasn't effective in reducing SHAPY shape variance. From these results, we can see that only averaging multiple different images works well on reducing variance from SHAPY and generating occluded images doesn't work well with it.

4.13 Analysis of the effect of averaging shape parameters

4.13.1 Visualize differences in individual shape parameters between gallery and query when using multiple images

For each person in 18 people in HBW, we calculate differences in shape parameters of meshes in the gallery and query set. We repeat this for one image (one in gallery, one in query), two images, ..., and the maximum number of images we have for that person.

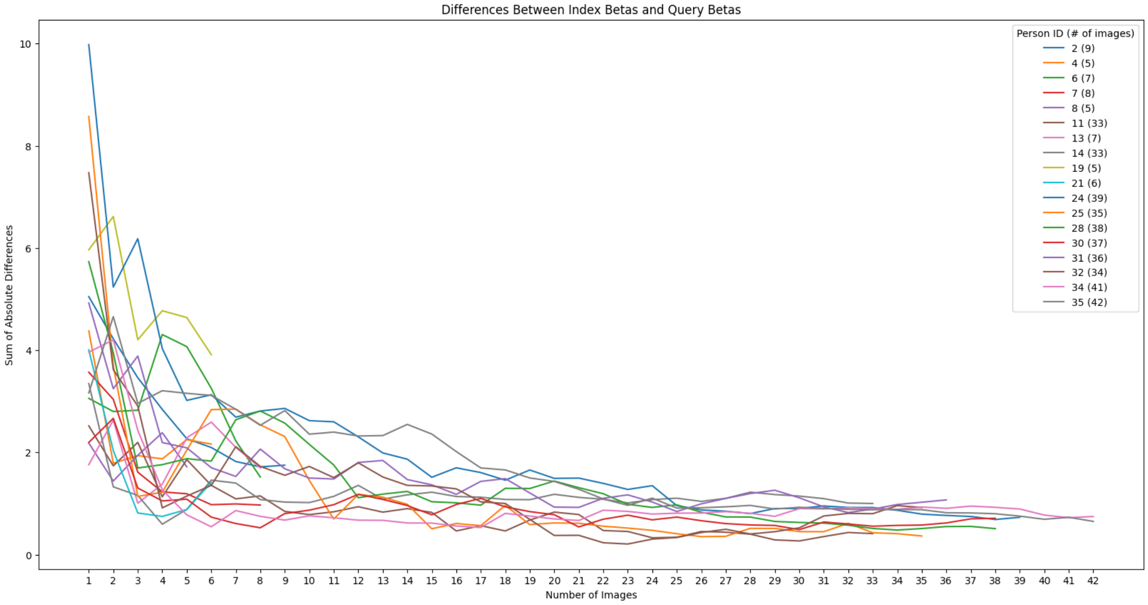


Figure 4.17: Differences in shape parameters between gallery and query when using multiple images

Figure 4.17 shows that when we use multiple images, the shape parameters of meshes in the gallery and query for each person close to each other.

4.13.2 Quantify differences in individual shape parameters between gallery and query when using multiple images

For each person, we calculate if we use two images (two for gallery, two for query), how much percent the differences of two averaging shape parameters will reduce compared to differences of shape parameters for a single case. We repeat 3 images, 4 images, ..., the maximum number of images we have for that person. Then for each ”number of images”, we average the percent results of all 18 people. (For example, for the number of images = 2, we calculate that percent value for person A, person B, ... person Z. Then I average

1 image	4 images	9 images	16 images	25 images	36 images	38 images
0%	47.41%	63.90%	74.20%	81.77%	80.65%	83.89 %

Table 4.10: Percent reduced of differences in individual shape parameters between gallery and query when using multiple images compared to using a single image

these percent values for all people to get the average percent of shape parameters reduced when using two images compare to use one image). We got results at Table 4.10.

These quantitative results surprise us. We can see that:

- When we use 04 images (04 galleries, 04 queries), the difference of shape parameters between gallery and query is reduced by 47.40% compared to using a single image (1 gallery, 1 query), roughly reduced to 1/2.

- When we use 09 images (09 galleries, 09 queries), the difference of shape parameters between gallery and query is reduced by 63.89% compared to using a single image (1 gallery, 1 query), roughly reduced to 1/3.

- When we use 16 images (16 galleries, 16 queries), the difference of shape parameters between gallery and query is reduced by 74.20% compared to using a single image (1 gallery, 1 query), roughly reduced to 1/4.

- When we use 25 images (25 galleries, 25 queries), the difference of shape parameters between gallery and query is reduced by 81.77% compared to using a single image (1 gallery, 1 query), roughly reduced to 1/5.

- When we use 36 images (36 galleries, 36 queries), the difference of shape parameters between gallery and query is reduced by 80.65% (at 38 is 83. 89%) compared to using a single image (1 gallery, 1 query), roughly reduced to 1/6.

These quantitative results are the same as we expected and are strong proof for our averaging shape parameters method.

4.14 Evaluate SHAPY 3D mesh virtual measurements on self-images

We also wanted to test the variance shape parameters that SHAPY predicts, so we took 9 pictures of one person, got SHAPY virtual measurements, and then compared them to his ground truth measurements.

Figure 4.18 shows several of our images used for this evaluation, also with SHAPY 3D model fit results. Table 4.18 showed the mean error of virtual measurements on self-images and on the MMTS dataset from the SHAPY paper. As you can see from the results from the table, self-images have a really high mean error on height and chest, while MMTS images have a high mean error on waist and hips. Both of these images have around 60mm to 100mm mean error. This error can be normal or acceptable in many areas like virtual reality (VR), augmented reality (AR), or any animation application, and even if it is state-of-the-art compared to previously 3D regressing methods, it is not acceptable in the field of human recognition when a small different prediction of two human bodies in two images can lead to incorrect recognition results. Due to this high error, we cannot get high results from multiple experiments before.

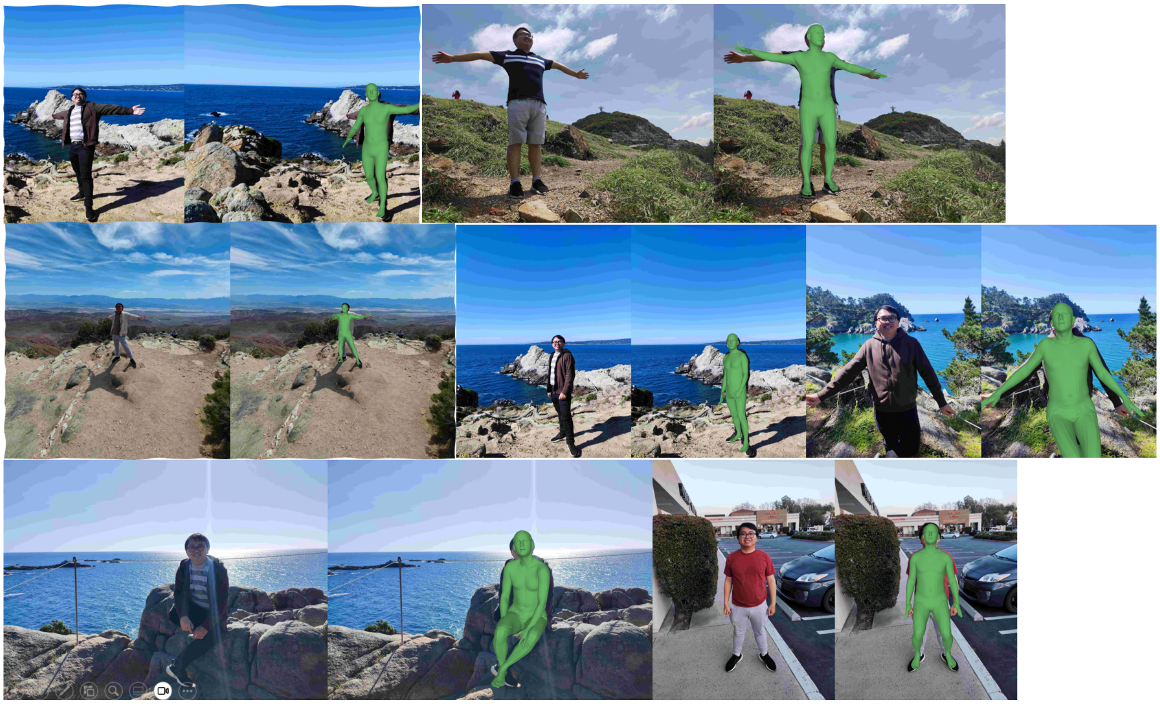


Figure 4.18: Several of our images and SHAPY fit results are used for virtual measurement evaluation

Mean error on	Mass (kg)	Height (mm)	Chest (mm)	Waist (mm)	Hips (mm)
Self images	2	123	67	**66**	**30**
MMTS		**71**	**64**	98	74

Table 4.11: Mean error of virtual measurements on self-images and on MMTS dataset from SHAPY paper

Chapter 5

Conclusion

Methods that regress the 3D human body from a single RGB image are improving day by day. SHAPY, the state-of-the-art method right now, is trained to regress more accurate human body shapes from images than previous methods with additional datasets by using two different ways to collect proxy annotations for 3D body shapes for in-the-wild images. Even though it has impressive results, it can predict two hugely different shapes of one person in two different images. Even with the same image but with the head occluded, SHAPY can produce different shape parameters. This error can be normal or acceptable in many areas like virtual reality (VR), augmented reality (AR), or any animation application, it is not acceptable in the field of human recognition when a small different prediction of two human bodies in two images can lead to incorrect recognition results. This means the 3D human body from the state-of-the-art right now cannot be used for human recognition.

We also improved the LSP method and made it work with 3D mesh instead of the 2D range image in the original paper. With our experiments and visualization, and also by the relationship between ranking and differences in shape parameters, LSP can capture the difference in body shape, and reflect the incorrect shape output of SHAPY. Recognition in 3D can be performed even in occlude images because SHAPY or many methods regress the full 3D body. If these methods can capture shape parameters consistently in these cases, LSP can totally have high accuracy with occluded images.

By averaging shape parameters of multiple images in both the gallery and query set, we have higher results than using shape parameters from only a single image because it will reduce the error of SHAPY shape prediction. We also experimented with augmentation combined with averaging shape parameters but it didn't have high results. From these results, we can see that only averaging multiple different images works well on reducing variance from SHAPY and generating occluded images doesn't work well with it.

Our research opens a way to recognize humans or classification with 3D objects. In the future, if we can somehow get a more accurate 3D human body for a person, either a new better method for regressing a 3D human body from a single RGB image or a scanned human body, we can totally use our LSP method to recognize human.

Chapter 6

Future Work

A pivotal aspect of our future work will involve delving into the underlying causes of variance in the shape parameters generated by SHAPY. This investigation will seek to identify specific factors that contribute to inconsistencies in shape representation, such as variations in input data quality, algorithmic biases, or limitations inherent in the model's design. Understanding these factors is crucial for improving the accuracy and reliability of shape parameter estimations.

Beyond SHAPY, our research will extend to examining alternative methodologies for 3D HPS (like HMR, SPIN, ...) to determine if they exhibit similar variances. This comparative analysis will involve evaluating their performance and assessing their robustness against variations in input data. The goal is to benchmark SHAPY against these alternatives to identify best practices and potential improvements.

To enhance the recognition results obtained from SHAPY, we will investigate a range of strategies aimed at reducing the variance of shape parameters. This will include the development and testing of new algorithms or modifications to existing ones, with a focus on improving parameter stability and accuracy. The effectiveness of these methods will be rigorously evaluated to identify the most promising approaches for enhancing SHAPY's output.

In an effort to achieve more consistent parameter outputs from SHAPY, we will explore the use of data augmentation and contrastive learning techniques. By fine-tuning SHAPY with a rich set of augmented data, we aim to improve the model's ability to generalize across different instances of the same subject, thereby reducing output variability and enhancing recognition performance.

The application of invariant representation methods, such as Domain-Adversarial Neural Networks (DANN), will be explored to achieve shape invariance across different poses. This approach aims to generate embeddings that are robust to variations in pose, thereby improving the consistency and reliability of shape parameter estimation.

Our future work will also investigate the possibility of segmenting the latent space embedding in a manner that isolates representations related to specific aspects of the shape. This targeted approach is anticipated to enhance the precision of shape parameter estimation by reducing the influence of extraneous factors, leading to more accurate and consistent shape representations.

Bibliography

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, jul 2005.

[2] Alexandru O. Balan, Leonid Sigal, Michael J. Black, James E. Davis, and Horst W. Haussecker. Detailed human shape and pose from images. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image, 2016.

[4] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007.

[5] Hui Chen and Bir Bhanu. Human ear recognition in 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):718–737, 2007.

[6] Vasileios Choutas, Lea Müller, Chun-Hao P. Huang, Siyu Tang, Dimitrios Tzionas, and Michael J. Black. Accurate 3d body shape regression using metric and semantic attributes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention, 2020.

[8] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Kosecka, and Ziyan Wu. Hierarchical kinematic human mesh recovery, 2020.

[9] Peng Guan, Alexander Weiss, Alexandru O. Bǎlan, and Michael J. Black. Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388, 2009.

[10] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12748–12757, 2021.

[11] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image, 2020.

[12] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies, 2018.

[13] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose, 2018.

[14] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation, 2020.

[15] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop, 2019.

[16] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on $\mathcal{X}$-transformed points, 2018.

[17] Junbang Liang and Ming C. Lin. Shape-aware human pose and shape reconstruction using multi-view images, 2019.

[18] Feng Liu, Minchul Kim, ZiAng Gu, Anil Jain, and Xiaoming Liu. Learning clothing and pose invariant 3d shape representation for long-term person re-identification. pages 19560–19569, 10 2023.

[19] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), oct 2015.

[20] Lea Müller, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black. On self-contact and human pose, 2021.

[21] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. *STAR: Sparse Trained Articulated Human Body Regressor*, page 598–613. Springer International Publishing, 2020.

[22] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image, 2019.

[23] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[24] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

[25] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization, 2020.

[26] Igor Santesteban, Elena Garces, Miguel A. Otaduy, and Dan Casas. Softsmpl: Data-driven modeling of nonlinear soft-tissue dynamics for parametric humans. *Computer Graphics Forum*, 39(2):65–75, May 2020.

[27] Mauricio Pamplona Segundo, Cole Hill, and Sudeep Sarkar. Long range gait matching using 3d body fitting with gait-specific motion constraints. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 603–612, 2023.

[28] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Synthetic training for accurate 3d human pose and shape estimation in the wild, 2020.

[29] Aravind Sundaresan, Brian Burns, Indranil Sur, Yi Yao, Xiao Lin, and Sujeong Kim. Human body model based id using shape and pose parameters, 2023.

[30] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes, 2018.

[31] Chien-Yao Wang and Hong-Yuan Mark Liao. YOLOv9: Learning what you want to learn using programmable gradient information. 2024.

[32] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[33] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. Icon: Implicit clothed humans obtained from normals, 2022.

[34] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6183–6192, 2020.

[35] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, Bill Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows, 2020.

[36] Haidong Zhu, Wanrong Zheng, Zhaoheng Zheng, and Ram Nevatia. Sharc: Shape and appearance recognition for person identification in-the-wild, 2023.