Utilizing Prior Knowledge for Non-Rigid Shape Completion

Omer Ben-Hayun
Supervised by Ido Imanuel
Electrical Engineering Department
The Technion - Israel Institute of Technology
Haifa 32000, Israel

March 10, 2022

Abstract

In recent years, researchers have shown an increased interest in 3D human pose and shape estimation. Most studies in the field relies solely on completion from partial shape without additional information, resulting a limited models that cannot always reconstruct the partial shape precisely. The study utilized prior based approach for shape reconstruction of human partial scans that significantly improved the performance of existing methods. Additionally, in this study we developed and applied new technique for sampling from large datasets resulting solid increase of the performance across all tested learning models. The sampling methodology presented here has profound implications for future studies of machine-learning models that relies on learning from large datasets. Finally, we designed new visualization tools to explore the shape and the pose manifold of parametric body models and datasets.

Keywords: 3D shape completion, Non-rigid geometry, FPS sampling, Single View Reconstruction.

1 Introduction

In recent years, major advances in computational capabilities have arise a growing demand for creating and consuming 3D content. However, professional scanning devices are too expensive to be used for the typical user. As a result, acquisition of 3D visual content is often limited by affordable depth sensors which use few number viewpoints and resulting substantial partiality of the complete shape. Thus, 3D shape completion plays a crucial role in addressing the issue of incomplete scans.

The required accuracy of the reconstruction can vary depending on the desired Application. For instance, in rigid scenarios like collision-free motion planning or robotic fruit-harvesting, only rough approximation is needed, and can exploit properties of Symmetry and context [9, 23]. In contrast, in many types of non-rigid cases like on the entertainment [15] or the medical imaging fields [1] accurate estimation is required. That is, precision is a key requirement in the completion process and the offered completion should respect the geometry of the original partial shape. Therefore, they often cannot be based solely on symmetry properties due to their non-rigid nature, and should be established upon more data in addition to the original scan.

[11] lists two basic approaches currently being adopted in research into shape completion of non-rigid shapes. One is generative based method and the second is alignment based method. Generative based approaches learn to approximate the class distribution and achieved impressive results in shape completion tasks. Yet, they suffer from notable methodological weaknesses, i.e. they are limited in that they only considers the partial shape during the completion time and does not take into account additional information that derived from the object. Hence, they failed to demonstrate generalization capabilities and cannot provide a accurate completion for unseen partial shapes. On the other hand, alignment based methods aiming to fit a complete shape to a partial shape. Since they exploit additional data during the inference time, they have potential advantage in terms of generalization and precise completions. However, current alignment based methods can carry only moderate partiality and considered to be slow.

This study set out to shine new light on shape completion tasks from several angles:

1. We introduce the design of visualization tools to explore the shape manifold of parametric body models.

2. We show a new methodology for choosing samples from large datasets that increase the performance of the learning process.

3. We propose a new architecture for shape completion from single partial view and another complete view in another pose.

4. We show a new architecture for shape completion from single complete view and another set of multiple partial views in another pose.

2 Related work

deep learning on 3D point clouds

There are different approaches to represent a 3D object such as (i) point cloud,(ii) triangular mesh,(iii) voxel grids, (iv) set of projected views (in other words, group of 2D images).

Whereas it is common to work with regulated input format like images or volumes in deep learning, a major problem with the first two representations is that they are usually not regulated. In order to overcome this issue, most researchers has been transforming those representations to one of the last two representations before utilizing them as part of the dataset for deep net architectures. Yet, there are certain drawbacks associated with the use of this approach such as unwanted voluminous of the original data and quantization artifacts. A more direct approach to learning from 3D point clouds can be found in [21] studies who introduce the point cloud networks method that able to learn from the raw point clouds. The advantage of this particular method arise from the nature of point cloud as a simple representation. It allows to describe complex meshes while avoiding combinatorial irregularities. Another advantage of using this networks is that its relatively has small computational footprint and supplement fine tuning between accuracy and complexity. This paper will utilise [21] approach on point clouds data for all of the resons stated above.

3D shape completion of non-rigid objects

shape completion task describes how to create a full 3D models from a partial data that acquired from an object in the real world. Generally, it has become commonplace to distinguish rigid from non-rigid types of models. The target model can be classified as rigid if it capable to change its shape over time only with rigid transformations - translation,rotation or reflection. For instance a couch, stone, sculpture and a coin are few examples for rigid models. Nevertheless, If the model is capable of changing its shape with non-rigid transformation in a way that maintain the distances between the points of the model along the surface of the model, this quantity also known as geodesic distances. In another words, non-rigid transformations also can be called isometries (e.g. bending). For example a robotic arm,human hand or a can all be classified as a non-rigid shapes. In this paper, we like to focus on 3D shape completion of non-rigid human scans.

Parametric models

3D models of the human body are one of the key components of 3D human body shape and pose estimation. Whereas there is a large number of published studies [3, 4, 12, 8, 13, 7, 20] that offered methods for human body models, SMPL model [14] can be considered as the most popular and very frequently used by the industry and the research community. It can accurately express a diverse human shape in different natural human poses. Together with SMPL compatible complementary body models: (i) MANO hands model [22] and (ii) DMPL soft tissues deformations model [14], the model encodes the 3D human body into shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{16}$ and pose parameters $\vec{\boldsymbol{\theta}}$ and coined as SMPL+H. The pose parameters $\vec{\boldsymbol{\theta}}$ consist of the following vectors: $\vec{\boldsymbol{\theta}} = (\boldsymbol{\theta}_{\text{transformation}}, \boldsymbol{\theta}_{\text{rotation}}, \boldsymbol{\theta}_{\text{body pose}}, \boldsymbol{\theta}_{\text{hands pose}}, \boldsymbol{\theta}_{\text{soft tissues}}) \in \mathbb{R}^{3\times3\times63\times90\times8}$ The body model decode the parameters tuple $(\vec{\boldsymbol{\theta}}, \boldsymbol{\beta})$ into human triangular mesh with $N = 6890$ vertices with the function $\mathcal{P}(\vec{\boldsymbol{\theta}}, \boldsymbol{\beta}) = \mathbb{R}^{3N}$. This paper contribute to the understanding of the pose and shape manifolds of datasets that use this body model with new visualization tools.

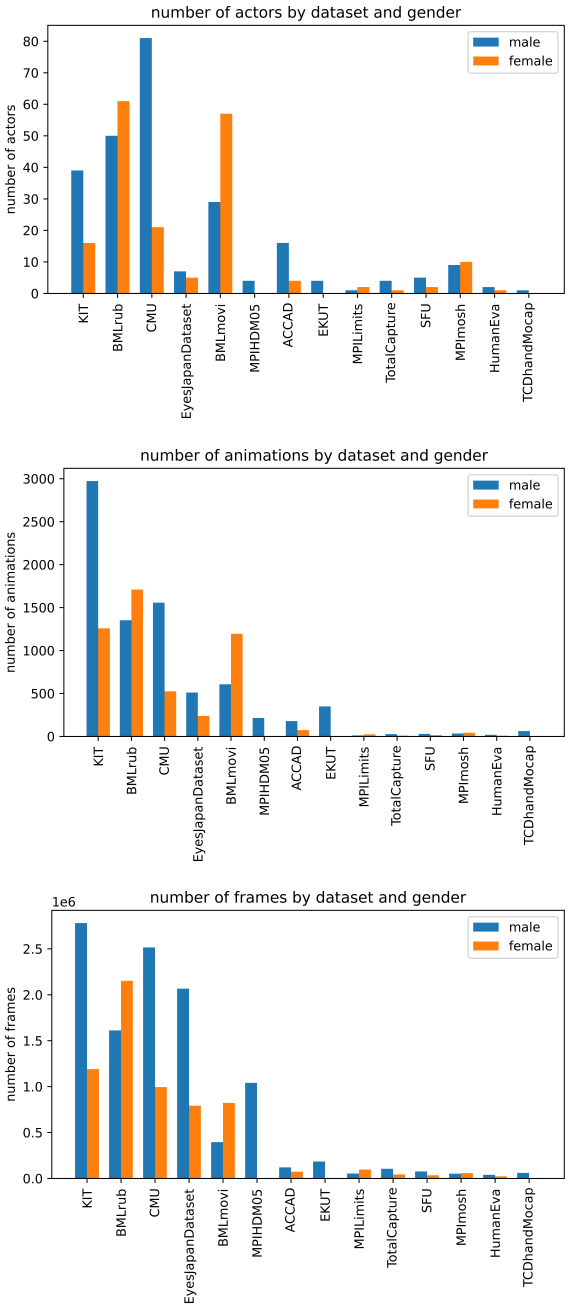

Figure 1: AMASS dataset unified from 14 external mocap datasets with varying number of actors, animations and frames.

AMASS dataset

By far, the most comprehensive dataset of human shapes is AMASS [17] dataset. AMASS merge 14 disparate archived datasets [Figure 1] to an aligned marker-based optical motion capture (mocap) data based on the SMPL+H body model. Each dataset in AMASS consist of actors and for each actor there is a group of animations that comprised from frames. In this respect, each actor have have its own shape parameters

$\boldsymbol{\beta}$ and for each frame can be describe as a pose parameters $\vec{\boldsymbol{\theta}}$. Throughout this paper, the term 'sample' encompasses the shape and pose parameters tuple $(\vec{\boldsymbol{\theta}}, \boldsymbol{\beta})$ that taken from the AMASS frame with its corresponding actor.

3 Method

3.1 Visualization tools

In the following paragraphs,we will briefly discuss two visualization approaches that we implemented in order to demonstrate and understand the amass body model and the pose manifold on AMASS dataset.

AMASS body model visualizer

This tool [Figure 2] was designed in order to examine the human body model parameterization used in AMASS. Namely to interactively view SMPL+H, $\mathcal{P}(\vec{\boldsymbol{\theta}}, \boldsymbol{\beta}) = \mathbb{R}^{3N}$. The user can interact with a GUI and choose which animation file to investigate. Moreover, The user can play the animation forward and backward, changing the original gender of the actor, and change the shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{16}$ and pose parameters $\vec{\boldsymbol{\theta}}$ for each frame in the animation and watch the resulting model in real time. The aim of this tool was to develop intuition about the human model parameters importance.

Figure 2: Screenshots from AMASS body model visualizer tool. The user can manipulate each element in the parameters tuple $(\vec{\boldsymbol{\theta}}, \boldsymbol{\beta})$ on the SMPL+H body model and immediately see the results. For illustrative purpose in this figure we limit some of the parameters dimensionality.

AMASS pose manifold explorer

Although AMASS have large number of samples, it is limited by the fact that it relies on multiple datasets with different quantities. For example, each dataset contain different samples in terms of motion complexities and variations. In other words, the variance in the pose manifold for each actor can be highly different. The aim of this tool is to visually explore the shape manifold for each actor and animation in AMASS [Figure 3]. In order to gain insights into the shape manifold variations for each actor the following steps were taken: First, we used principal component analysis (PCA) for dimensionality reduction of each sample $\boldsymbol{\theta}_{\text{body pose}} \in \mathbb{R}^{63}$ into a reduced 3D version $\hat{\boldsymbol{\theta}}_{\text{body pose}} \in \mathbb{R}^{3}$. Secondly, we group all the 3D projected pose samples for each actor and plot them in space colored acording to the animation and the frame. Thirdly, for each selected frame we display the corresponding pose on a natural template actor shape. Finally, we added 3 sliders that controls the current actor,animation and the frame.

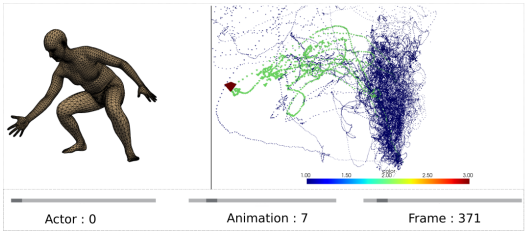

Figure 3: AMASS pose manifold visualizer. The big red point is the current frame sample, the green points represents the current animation and the remaining blue samples are the reset of the frames for the actor.

3.2 Sampling the dataset

In order to address the variations in the pose manifold variance for AMASS dataset [Figure 5] and the pose bias for each actor [Figure 4], we develop a new methodology for choosing samples based on the samples distances. The first step in this process was to choose the 10 female and 10 male actors with the highest variance on AMASS and dividing them to group based on gender. Afterwards, from each actor we sampled two kinds of frame sets with 50 samples. Whereas The first set was chosen randomly, the second was sampled using the Farthest point sampling (FPS) [Algorithm 1]. It aims to sample a subset of points that are farthest away from each other, resulting a subset of samples that are unbiased in terms of pose manifold [Figure 6]. Finally, we assembled together all the combinations each gender with each set type, resulting 4 different datasets [Table 1].

3.3 Deep learning models

Each shape will be represented as a point cloud embedded in $\mathbb{R}^3$. Generally, each point can represent another types of data in addition to it's 3D coordinates such that each point embedded in $\mathbb{R}^d$, but for this formulation we will use $d = 3$. Our objective is, given a full shape $Q = \{q_i\}_{i=1}^{n_q}$ and a target partial shape in different pose $P = \{p_i\}_{i=1}^{n_p}$ to reconstruct

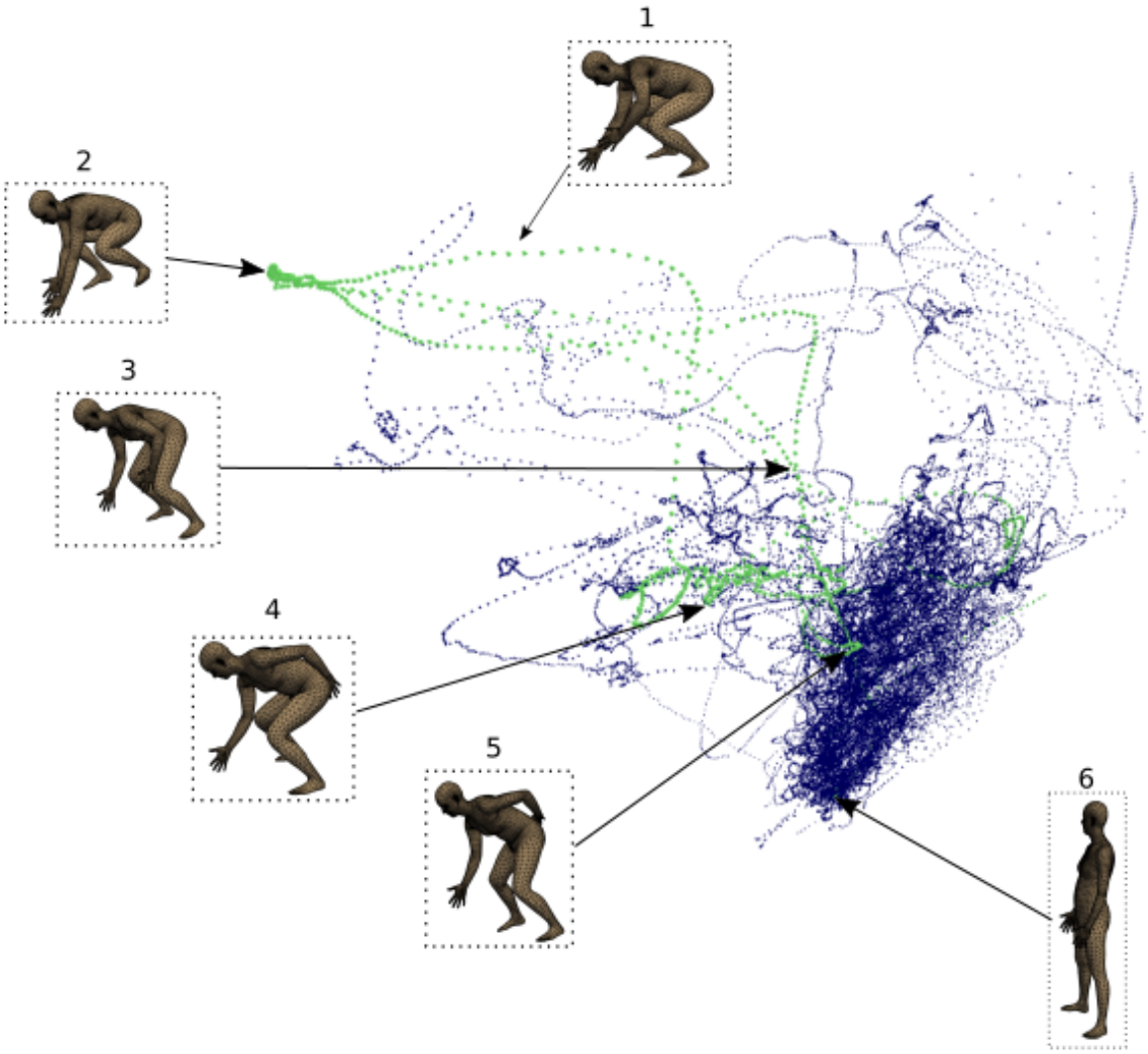

Figure 4: 3D PCA projection of the pose manifold of an actor for series of 6 chronological frames of the same animation. What is interesting in this figure is the general pattern of the animation continuity. In other words, closer frames in the time domain reflected on the projected domain by short distances between the samples. Perhaps one of the most important finding from those projections are the massive centroids that are presents poses that are closer to the common rest poses for the actor. In this respect, it can be seen from those centroids that the pose manifold of the different actors is biased towards the rest poses.

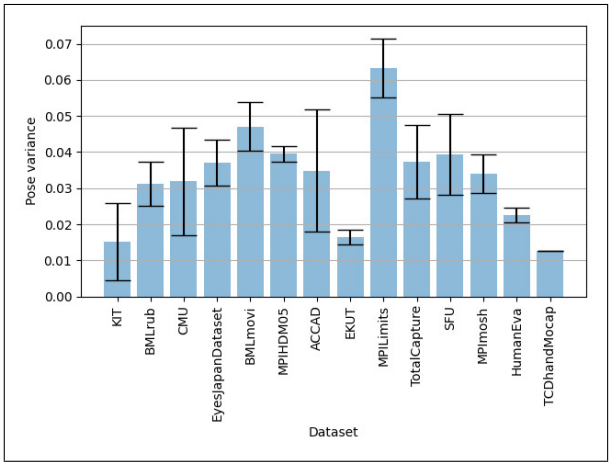

Figure 5: The pose parameters variance Var($\boldsymbol{\theta}_{\text{body pose}}$) for each dataset in AMASS. What stads out in this chart is the variability between different datasets. For example, in MPI_LIMITS dataset [2] includes an wide-ranging variety of human poses in contrast to EKUT dataset [18] the contain smaller variability in the pose manifold.

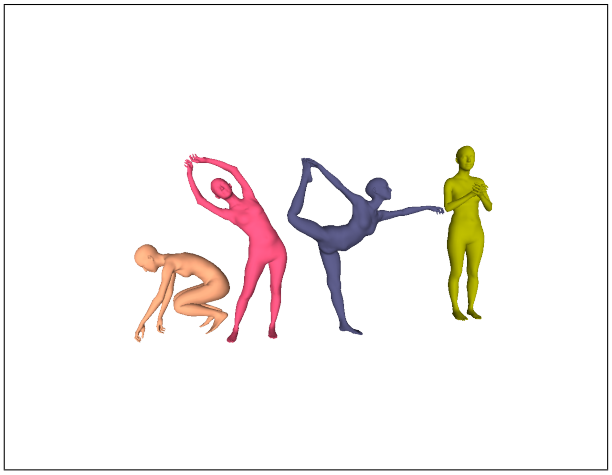

Figure 6: The first 4 poses that sampled for a single female actor on BMLmovi[24]. What can be clearly seen in this figure is the high variability of the poses achieved using FPS sampling.

Algorithm 1: FARTHEST POINT SAMPLING (FPS)

Data:
$N \in \mathbb{N}^+$ the number of frames of a given actor.
$n \in \{1, 2, ..., N\}$ the number of frames to be sampled.
$\boldsymbol{\Theta}^{\text{body pose}} = \{\boldsymbol{\theta}^{\text{body pose}}\}_{i=1}^{N}$ the body pose vectors for each frame.
Result: selected frames $S \subseteq \{1, 2, ..., N\}$, $|S| = n$
Function *FPS*(N , n , $\boldsymbol{\Theta}^{body\ pose}$)
  if n=N **then**
    $S \leftarrow \{1, 2, ..., N\}$
    return S
  sample randomly: $s \leftarrow \mathcal{U}\{1, 2, ..., N\}$
  $S \leftarrow \{s\}$
  $U \leftarrow \{1, 2, ..., N\} \setminus S$
  while $|S| < n$ **do**
    $\forall i \in U : d_i^{min} =$
    $\min_{j \in U; j \neq i} \left|\left| \boldsymbol{\theta}_j^{\text{body pose}} - \boldsymbol{\theta}_i^{\text{body pose}} \right|\right|_2$
    $s \leftarrow \arg\max_{i \in U} d_i^{min}$
    $S \leftarrow S \cup \{s\}$
    $U \leftarrow U \setminus \{s\}$
  return S

	Males	Females
Random	high-variance males random (MR)	high-variance females random (FR)
FPS	high-variance males fps (MF)	high-variance females fps (FF)

Table 1: Different sampling methods datasets names

P to its full shape $R = \{r_i\}_{i=1}^{n_r}$. In other terms, we interested to find R that is close as possible to the ground truth unknown full shape $G = \{g_i\}_{i=1}^{n_g}$ that P was acquired from. In order to archive this goal, we are are using a fixed template $T = \{t_i\}_{i=1}^{n_t}$ of a full shape in the "zero" pose [Figure 9] and try to learn a deformation function $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ such that $F(T) = R$. From this prospective, we can say that θ is the deformation parameter for the fixed template T, also known as the global shape descriptor. It is important to note that because we are using a body model with fixed vertices number N we know that , $N = n_t = n_r = n_q = n_g$, particularly $n_t = n_r$ and therefore F is well defined. It is, of course, important to acknowledge that input pair (Q, P) will influence the reconstruction function F. Therefore, we denote θ as a latent encoding of the input pair (Q, P) and model this correlation by offering a parametric function $F_\theta : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. The implementation of this process using a neural-network of encoder and decoder tries to learn the space of deformations θ as well as the decoder and encoder weights in order to archive a precise completion of P. The architectural approach taken in this study is a mixed architecture based on two papers. Firstly, the deformation of a fixed template as shown in 3D-CODED [10] and secondly, the usage of an input full prior shape from another pose as seen on [11].

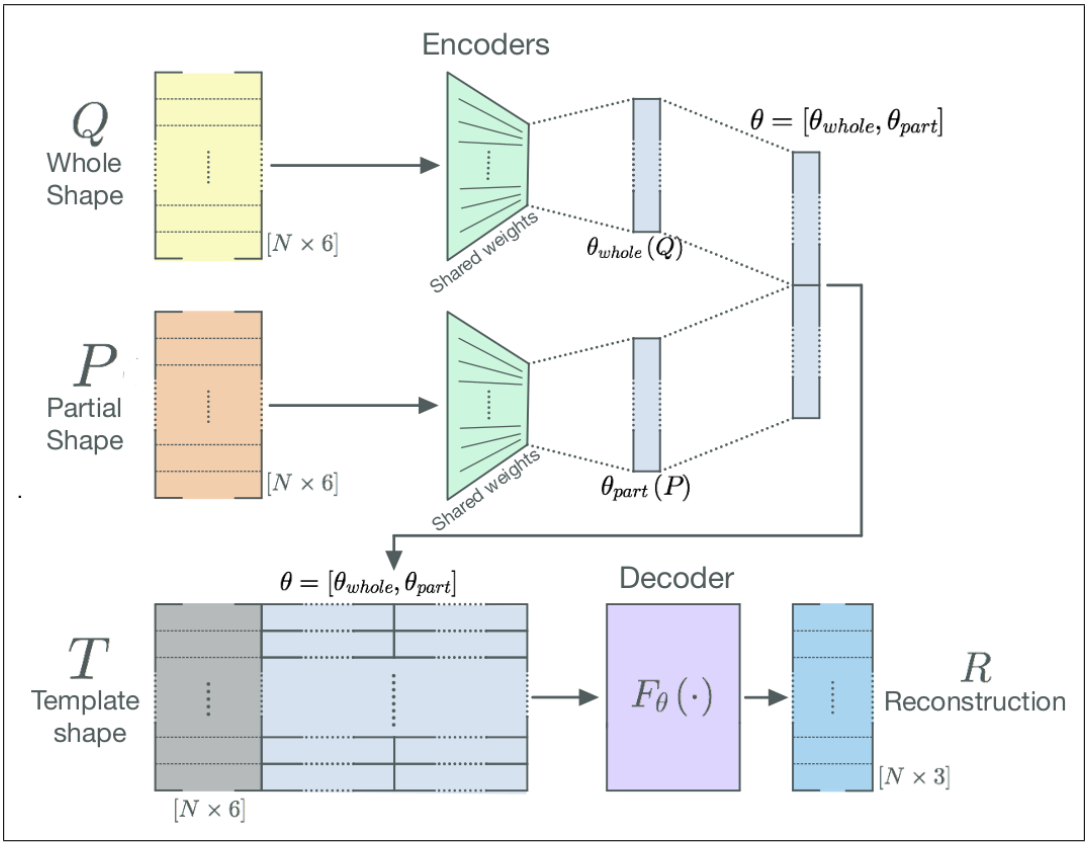

Figure 7: **FTP Architecture**. The top help of the figure describe the encoding process while the decoding phase is presented at the bottom. **Completion process:** The first step in this process is to provide the partial shape P and the whole shape prior Q into the encodes. Each shape is consist of N 6D points that comprised of coordinates and its corresponding unit normal vector. The encoded shapes, $\theta_{\text{whole}}(Q), \theta_{\text{part}}(P)$ then concatenate together, resulting the full global shape descriptor θ. Prior to the decoding phase, we concatenate the latent code of the input shape θ for all the 6D points on T.Notice that θ is the same across all the points in T. Finally, to obtain the full reconstruction R, we calculate the deformation for the template shape $F_\theta(T) = R$.

Encoder

Based on PointNet [21] architecture and [11] ideas, our encoder is comprised of two single-shape encodes. Each single-shape encoder take a point cloud and encode it to a global shape descriptor θ. That is, given the input tuple (Q, P) , the encoders produce the corresponding tuple $(\theta_{\text{whole}}, \theta_{\text{part}})$. In the final stage of the encoder, we concatenate the output tuple to the final latent space partial shape encoding $\theta = [\theta_{\text{whole}}, \theta_{\text{part}}]$. As proposed in [11], we added for each point its normal vector that computed using the connectivity for the underlying input mesh, therefore each point in the input is 6D.

Generator

Once the global shape descriptor θ was extracted from the input tuple (Q, P), the parametric deformation function F_θ will be predicted by the generator. After F_θ prediction, we append θ for each point t_i of the fixed template, resulting the generator input tuple $t_i^\theta = (t_i, \theta)$. It is important to bear in mind that θ remain fixed throughout all the generator input tuples. Finally, for each point we compute the prediction full shape $r_i = F_\theta(t_i^\theta) \in \mathbb{R}^3$ Furthermore, if we want to compute the unit normal vectors for 6D point cloud output, we can either calculate it with a known connectivity of the triangular mesh or predict it using differential normal estimators [21, 5].

loss functions

In addition, The loss function must represent the accuracy of the offered reconstruction. However, we will use a simple calculation [Equation 1] that based on the Euclidean proximity between the reconstruction R and the ground truth G.

$$\mathcal{L}(R, G) = \sum_{i=1}^{N} \left|\left| g_i - r_i \right|\right|_2^2, \tag{1}$$

To calculate the loss function, we assume that the correspondence $r_i \leftrightarrow g_i$ is known, and $r_i \in R$ is the matched point of $g_i \in G$. It has been demonstrated [11] that the concatenation of the normal vectors for the points can increase the fine

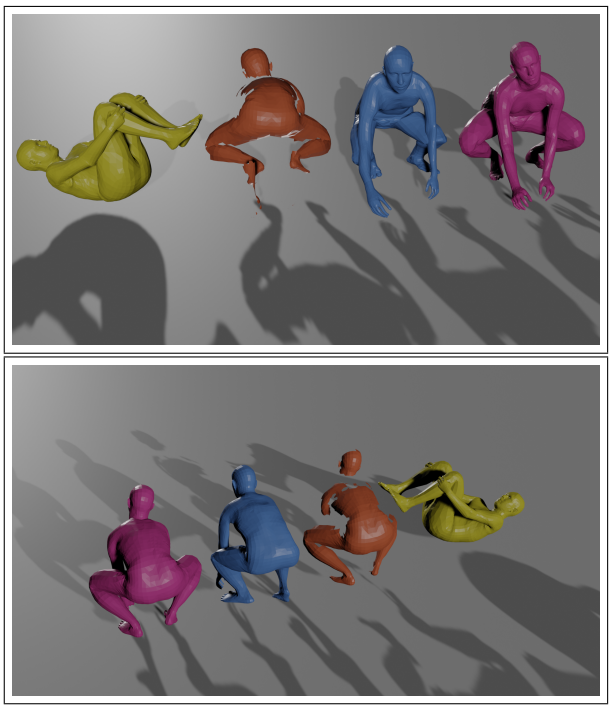

Figure 8: Example for input tuple and the resulting completion from diffrent angels. **Yellow** input reference shape Q, **Orange** partial shape P, **blue** FTP completion R, **Pink** ground truth G.

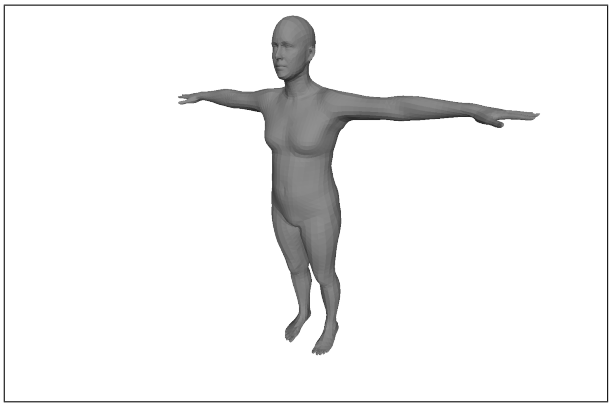

Figure 9: The template shape used as T. This template is corresponding to the "zero" shape of SMPL+H body shape model, namely $(\vec{\boldsymbol{\theta}}, \boldsymbol{\beta}) = (\vec{\mathbf{0}}, \mathbf{0})$.

details of the reconstruction. As discussed above, we concatenate the unit normal vectors to its coordinates, resulting a 6D points on the output, namely $g_i, r_i \in \mathbb{R}^6$.

Another architectural variations

The architecture described above will be coined here as **Fixed template with prior (FTP)**. In order to compare it past architectures we will briefly describe them here.

Fixed template without prior (3D-CODED) This variant is an implementation of the architecture that appear on 3D-CODED [10]. Whereas FTP is an input full prior shape from another pose Q, on 3D-CODED [10] we use only a single shape encoder for the partial shape P, Thus, on this variant the global shape descriptor $\theta = [\theta_{\text{part}}]$.

changing template with prior (Towards precise) This variant is the original implementation of the architecture that appear on 'Towards-precise' [11] method. While FTP is an concatenate the global shape descriptor θ to the fixed template points T, on Towards-precise[11] we will use the full shape from different pose Q points instead. Formally, the reconstruction process can be described as $r_i = F_\theta(q_i^\theta) \in \mathbb{R}^3$ where, of course, $q_i^\theta = (q_i, \theta)$.

Furthermore, we will offered another variation of the architecture that aims to complete a partial shape from multiple partial point clouds given as priors that will be evaluated separately.

Fixed template with N multiple priors (FTMP) This variant aggregates multiple nonrigid views of the same person as the prior with multiple shape encoders. More precisely, instead of using full shape from another pose Q as our prior this version will take N **partial** point clouds $A = \bigcup_{j=1}^{N} A_j$ and encode all of them together on shape encoder with shared weights, resulting θ_A. After the concatenation, the final shape descriptor is $\theta = [\theta_{\text{part}}, \theta_A]$.

Implementation details

All of our neutral networks was trained with PyTorch [19] together with ADAM optimizer that configured with momentum of 0.9 and a constant learning rate of $10^{-}3$. As mention above, each shape contained $N = 6890$ 6D points. Moreover, the shape-descriptor sizes were same as [11], namely $|\theta_{\text{part}}| = |\theta_A| = |\theta_{\text{whole}}| = 512$. As shown in [11] and mentioned above we concatenate the scaled unit normal vectors to the coordinates for each point $s_i = (\vec{x}_{si}, \alpha\vec{n}_{si}) \in \mathbb{R}^6$, with $\alpha = 0.1$.

In addition, each batch contained 10 shapes tuples (P, K, R). In this perspective, K is the prior element for each

variant:

$$K = \begin{cases} Q & \text{FTP (ours) or Towards precise [11]} \\ \emptyset & \text{3D-CODED [10]} \\ \{A_1, A_2, ..., A_N\} & \text{FTMP (ours)} \end{cases} \quad (2)$$

Finally, all the input shapes were centralized such that their corresponding full shapes center of mass is aligned with the origin.

4 Experiments

To compare the difference between all sampling method and the architectures, we trained all the single prior architectures, namely FTP (ours), Towards-precise [11] and 3D-CODED [10] on all of the different datasets that described on [Table 1]. Additionaly, we split the each dataset to be actor aligned, That is, the validation, test and train sets contained different actors. On other words, the test set comprised only from unseen actors that did not appear on the validation, nor the training set, and also all the actors that appear on the validation set were not included on the training set. Moreover, in order to test the sampling methods without pose manifold bias, we always used the FPS sampling method for the validation and test sets [Table 2]. Finally, we used 10K,1K,1K as our data split for the train, validation and test set respectively.

	Train	Validation	Test
males random	MR	MF	MF
males fps	MF	MF	MF
females random	FR	FF	FF
females fps	FF	FF	FF

Table 2: Different sampling methods datasets splits. First charterer refer to the dataset actors gender (namely, male or female) and the second to the sampling method e.g. Random or FPS.

4.1 Evaluation metrics

In attempt to evaluate each experiment performance, we calculated several metrics that demonstrated the completion quality. l_n mean point-wise distance errors [Equation 3] for $p \in \{1, 2, \infty\}$ refers to the mean of the l_n norm between each reconstruction point and its corresponding ground truth point. In this respect, each point is always 3D and represent the point coordinates in $\mathbb{R}^3$.

$$\forall p \in \{1, 2, \infty\} : \mathbb{E}_{l_n(G,R)} = \frac{1}{N} \sum_{i=1}^{n} \left|\left| g_i - r_i \right|\right|_p \quad (3)$$

Finally we compute the normalized quantities of the volume, surface area and surface area to volume ratio (denoted here as SVR) [Equation 4].

$$\forall O \in \{Vol, Area, SVR\} : M_O(G, R) = \frac{|O(G) - O(R)|}{O(G)} \cdot 100 \quad (4)$$

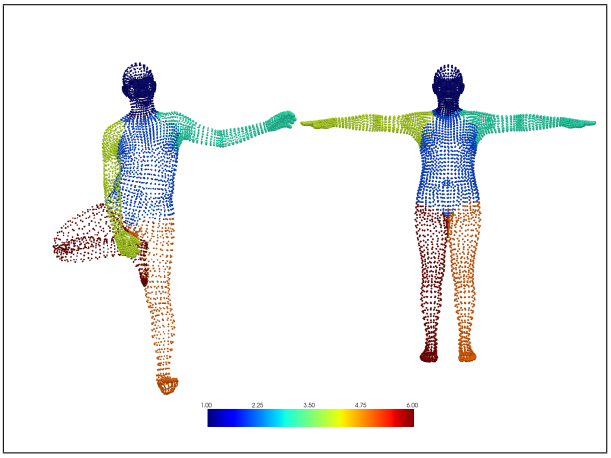

Figure 10: The diffrent segmentation elements colored on an arbitrary pose (left) and on the template model T (right). **Segments color coding** : head (1), torso (2), left arm (3), right arm (4), left leg(5), right leg (6).

To analysis the preciseness of each experiment in greater detail, we also computed all of the metrics of the current section on 6 different segments: (i) head, (ii) torso, (iii) left arm, (iv) right arm, (v) left leg, (vi) right leg [Figure 10]. In this respect, the volume computed after we closed each segment mesh by appending predetermined faces to each mesh. Moreover, the mean point-wise distance errors [Equation 3] average is taken over the number of points for each segment i.e.

$$N = \#\text{Points in segment}$$

FTMP evaluation metric

In order to evaluate the complexity for the completion task we provide vertex cover metrics that aim to determined the amount of information in each prior. As similar to the loss calculation, To calculate the vertex cover metrics , we assume that all the vertices correspondence is known. For N $\in \mathbb{N}^+$ prior point clouds, given the full prior $A = \bigcup_{j=1}^{N} A_j$ and the ground truth shape G, we can define the remaining unseen vertices as $\mathcal{R} = G \setminus A$. Therefore, the new vertices set the prior supplement is $\mathcal{N} = A \cap \mathcal{R}$.

- Vertex cover 1 can be defined as $\text{VC}_1 = \frac{|\mathcal{N}|}{|\mathcal{R}|} \in [0, 1]$. This definition represent the amount of new vertices on A in relation to the remaining unseen vertices.
- Vertex cover 2 can be defined as $\text{VC}_2 = \frac{|P \cup A|}{|G|} \in [0, 1]$. This definition demonstrate the new vertices on the full input, namely $P \cup A$ in relation to the full vertex set G.
- Vertex cover 3 is the arithmetical mean of the definitions above, i.e. $\text{VC}_3 = \frac{\text{VC}_1 + \text{VC}_2}{2} \in [0, 1]$.

On the special case that $N = 0$, we defined the $\text{VC}_1 = \text{VC}_2 = \text{VC}_3 = 0$.

4.2 Comparing sampling methods

$\mathbb{E}_{l_2}$ Error [cm]	FTP (ours)	3D-CODED [10]	Towards precise [11]
males random	0.0977	0.1142	0.1584
males fps (ours)	0.0390	0.0474	0.0908
imrprovement [%]	**250.5**	**240.9**	**174.4**
females random	0.0786	0.0715	0.1399
females fps (ours)	0.0348	0.0363	0.08244
imrprovement [%]	**225.86**	**196.96**	**169.69**

Table 3: l_2 mean point-wise distances error, i.e. $\mathbb{E}_{l_2}$ computed both on fps and random sampling techniques for male and female datasets [Table 2] across three shape completion methods.

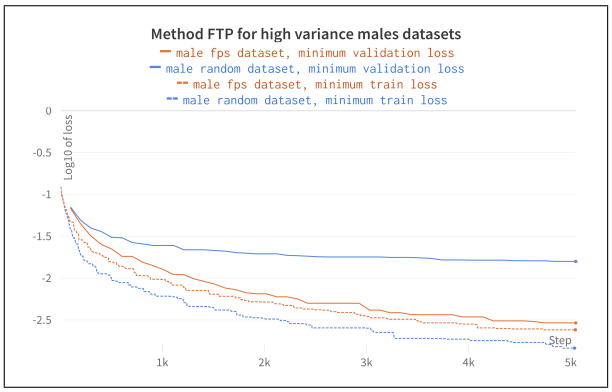

Figure 11: Comparison of different sampling methodologies in terms of training loss convergence for FTP method on high variance male datasets. The loss term is the addition of the minmum square error (MSE) of the coordinates and the MSE error of the corresponding normal vectors multiplied by 0.1

4.3 Shape reconstruction comparison

[Table 4] provides the results obtained from the analysis of the different shape completion methods on the high-variance males dataset.

4.4 FTMP multiple priors reconstruction

For this experiments we used a bigger variant of the AMASS [17] dataset: We choose all the actors that contained at least 10K frames from 5 different AMASS datasets: KIT [18], BMLrub [24], CMUa [6], EyesJapanDataset [16] and BMLmovi [24]. The final stage was to sample from each actor 1K samples using Farthest point sampling (FPS) [Algorithm 1]. [Figure 12] shows the summary statistics for our FTMP method for different types of prior cases for $N \in \{0, 1, ..., 8\}$ partial prior point clouds.

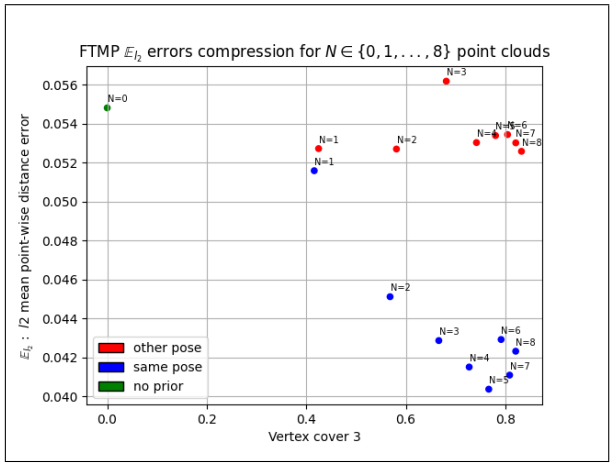

Figure 12: FTMP $\mathbb{E}_{l_\infty}$ error with respect to VC_3 and the number of point clouds N. The full prior $A = \bigcup_{j=1}^{N} A_j$ was obtained from the same actor as G but differ in two different scenarios. **Blue samples** represents the case that all partial point clouds in A taken from the same pose as P. **Red samples** describe the case that all the partial point clouds in A was taken from a specific pose that differ from G. **Green sample** represents the $N = 0$ case; in other words, in this case we do not use any prior $A = \emptyset$, this completion method is equivalent to 3D-CODED [10].

5 Discussion

5.1 Comparing sampling methods

The first set of analyses examined the impact of the FPS sampling methodology [Algorithm 1]. It can be seen from the data in [Table 3] the FPS sampling method increased the performance significantly across various shape completion models on both gender cases. What is striking about the results in this table is that the performance improvement is **at least** $\mathbf{150}\%$ across all cases. This is a remarkable result. Moreover, [Figure 11] indicate a major difference between the sampling methods during the training phase. While the model that train on random sampling method is over-fitting, the same model that trained on FPS sampling demonstrated much higher generalization capabilities. These interesting findings might be explained by the fact that the FPS sampling methodology resulting high variability pose space with small bias. It represents the neutral human pose space more accurately unlike the random sampling methodology that is keeping the underlying bias [Figure 4] of the original sampled dataset in the terms of pose manifold. In future investigations, it might be possible to use a different machine-learning scenarios like noise-reduction or image classification, in which this sampling methodology could increase the performance of existing models.

	Segment						
	full body	head	torso	left arm	right arm	left leg	right leg
$\mathbb{E}_{l_1}$ **Error [cm]**							
Towards precise	0.146208	0.102846	0.115495	0.144648	0.153006	0.193752	0.200084
3D-CODED	0.078359	0.067634	0.061843	0.074011	0.082207	0.096085	0.099974
FTP (ours)	**0.066677**	**0.054331**	**0.049950**	**0.066002**	**0.063711**	**0.091557**	**0.091371**
$\mathbb{E}_{l_2}$ **Error [cm]**							
Towards precise	0.098178	0.068907	0.077451	0.096660	0.102748	0.131054	0.134588
3D-CODED	0.052263	0.044773	0.041285	0.049226	0.055014	0.064134	0.067086
FTP (ours)	**0.044519**	**0.035908**	**0.033397**	**0.044215**	**0.042241**	**0.061283**	**0.061720**
$\mathbb{E}_{l_\infty}$ **Error [cm]**							
Towards precise	0.082236	0.057631	0.064776	0.080546	0.086051	0.110618	0.112889
3D-CODED	0.043444	0.036744	0.034373	0.040831	0.045964	0.053414	0.056166
FTP (ours)	**0.037056**	**0.029499**	**0.027834**	**0.036889**	**0.034887**	**0.051279**	**0.052105**
Volumetric error [%]							
Towards precise	30.423622	33.110760	25.966375	39.425831	41.843704	43.813766	41.952892
3D-CODED	**9.939109**	15.763901	**9.738126**	22.458927	27.767376	16.987247	18.243286
FTP (ours)	13.418878	**12.206391**	10.964217	**18.980408**	**24.697559**	**15.907495**	**17.564968**
Surface area error [%]							
Towards precise	19.716169	22.305054	15.056305	25.185989	28.640442	29.563946	28.295647
3D-CODED	**6.657650**	9.931189	**6.170079**	16.490198	19.626400	10.011135	11.033092
FTP (ours)	9.318895	**7.568487**	7.852102	**13.054072**	**16.515900**	**9.097844**	**10.734350**
Surface area to volum error [%]							
Towards precise	28.587654	23.371700	24.144958	36.686630	38.859291	37.980103	40.409191
3D-CODED	**4.852785**	7.017126	4.835610	9.741541	13.645969	**8.563823**	**8.851275**
FTP (ours)	5.461816	**5.458884**	**4.571476**	**9.236479**	**12.392942**	9.308583	9.566586

Table 4: Comparison of Towards-precise [11], 3D-CODED [10] and FTP shape completion methods with respect to the described evaluation metrics on each segment. All the methods was trained on the high-variance males fps (MF) dataset. The minimum value on each column appear in bold.

5.2 Shape reconstruction comparison

[Table 4] compare the results obtained from the shape reconstruction comparison experiments across all the segments and the entire shape. Each evaluation metric was carried out on each segment in addition to the whole shape and appear on separate panel.As can be seen from the table, Towards-precise [11] method have much inferior preformance across all the metrics and segments as well for the full shape, in comparison to reset of the method. A possible explanation for this result might be related to [11] architectural design that aims to find the dense correspondence between each full shape G and another arbitrary pose partial shape P while on the same time approximate the completion shape for P. As a result, it tries to solve much complecated problem than the original shape completion problem.Closer inspection of the table shows number of important differences between 3D-CODED [10] and our method. Crosswise all the mean point-wise distance errors [Equation 3] and throughout all the segmentations and the full shape comperation , our method constantly outpreform 3D-CODED method. Additionaly, there was unambiguous difference between the methods when considering normalized quantities [Equation 4] across the diffrent segments. When considering the volumetric error and the surface area error, 3D-CODED method achieved better results on the torso segment and on the full body. However, our method surpass 3D-CODED on the other segemts, e.g. head, arms and legs. Taken together, these findings suggest a role for prior partial shape P in promoting precie shape completion.

5.3 FTMP multiple priors reconstruction

[Figure 12] presents the breakdown of FTMP method $\mathbb{E}_{l_2}$ error according to number of point clouds N on the prior and to the origins of this prior. For the case that all partial point clouds in A taken from the same pose as G (which represented as blue samples), The scatter plot shows that there has been a steady decrease in the $\mathbb{E}_{l_2}$ error for each point for $N \in \{0, 1, ..., 5\}$. This is a somewhat reassuring result. However, increasing N further causing an unwanted outcome: the dimensionality of the input space is becoming bigger and as a result, the error stop to decrease. As a generalization attempt for FTP method, we also run the experiments with the scenario that A taken from the different pose from G (which represented as red samples). On this case, There was no significant difference between the experiments with respect to the error rate. It is difficult to explain this result, but it might be related to the usage of the size shape descriptor components sizes relation $\theta = [\theta_{\text{part}}, \theta_A]$. In our experiment, θ comprised of two vectors with same lengths $|\theta_{\text{part}}| = |\theta_A| = 512$. In future investigations of this scenario, it might be possible to use a different lengths for those components, such that $|\theta_{\text{part}}| > |\theta_A|$ in which reflect the importance of the target point could P related to the prior partial point clouds A. Therefore, it is possible that for this case, these results do not accurately reflect the true potential of the model and more research on this model needs to be undertaken.

6 Conclusions

The main aim of the present research was to examine methods for accurate 3D shape completion. This study set out to develop neural-network architecture for 3D prior based shape completion method and evaluate it compared to existing methods.The current results highlight that precise shape completion can be achieved through utilization of prior data. On this aspect, further studies are required to develop a deeper understanding of the prior based learning effectiveness in 3D shape completion field. The research has also shown that choosing samples from large datasets according to our new methodology significantly increase the performance of shape completion models. These findings have crucial implications for the understanding of how the variance of the training dataset could effect machine-learning models in various of applications. Finally, we provide two new visualization tools which provides new ways to explore the shape and the pose manifolds of parametric body models and datasets.

7 Acknowledgements

Most of all, I would like to thank Ido Imanuel, my supervisor, for his constant and continued support and patience. Ido has monitored my progress and offered advice and encouragement throughout. My gratitude is also extended to my sister Adi Ben-Hayun for helping me illustrate figure 7.

References

[1] Abdi, A. H.; Pesteie, M.; Prisman, E.; Abolmaesumi, P.; and Fels, S. 2019. Variational shape completion for virtual planning of jaw reconstructive surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 227–235. Springer.

[2] Akhter, I., and Black, M. J. 2015. Pose-conditioned joint angle limits for 3d human pose reconstruction. 1446–1455.

[3] Allen, B.; Curless, B.; and Popović, Z. 2002. Articulated body deformation from range scan data. *ACM Transactions on Graphics (TOG)* 21(3):612–619.

[4] Anguelov, D.; Srinivasan, P.; Koller, D.; Thrun, S.; Rodgers, J.; and Davis, J. 2005. Scape: shape completion and animation of people. 408–416.

[5] Ben-Shabat, Y.; Lindenbaum, M.; and Fischer, A. 2019. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks. 10112–10120.

[6] Carnegie Mellon University. CMU MoCap Dataset.

[7] Chen, Y.; Liu, Z.; and Zhang, Z. 2013. Tensor-based human body modeling. 105–112.

[8] Freifeld, O., and Black, M. J. 2012. Lie bodies: A manifold representation of 3d human shape. 1–14.

[9] Ge, Y.; Xiong, Y.; and From, P. J. 2020. Symmetry-based 3d shape completion for fruit localisation for harvesting robots. *Biosystems Engineering* 197:188–202.

[10] Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018. 3d-coded: 3d correspondences by deep deformation. 230–246.

[11] Halimi, O.; Imanuel, I.; Litany, O.; Trappolini, G.; Rodolà, E.; Guibas, L.; and Kimmel, R. 2020. Towards precise completion of deformable shapes. 359–377.

[12] Hasler, N.; Stoll, C.; Sunkel, M.; Rosenhahn, B.; and Seidel, H.-P. 2009. A statistical model of human pose and body shape. 28(2):337–346.

[13] Hirshberg, D. A.; Loper, M.; Rachlin, E.; and Black, M. J. 2012. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. 242–255.

[14] Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34(6):248:1–248:16.

[15] Lozada, R. M.; Escriba, L. R.; and Granja, F. T. M. 2018. Ms-kinect in the development of educational games for preschoolers. *International Journal of Learning Technology* 13(4):277–305.

[16] Ltd., E. J. C. Eyes Japan MoCap Dataset.

[17] Mahmood, N.; Ghorbani, N.; Troje, N. F.; Pons-Moll, G.; and Black, M. J. 2019. AMASS: Archive of motion capture as surface shapes. 5441–5450.

[18] Mandery, C.; Terlemez, .; Do, M.; Vahrenkamp, N.; and Asfour, T. 2015. The KIT whole-body human motion database. 329–336.

[19] Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

[20] Pons-Moll, G.; Romero, J.; Mahmood, N.; and Black, M. J. 2015. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)* 34(4):1–14.

[21] Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation.

[22] Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36(6).

[23] Schiebener, D.; Schmidt, A.; Vahrenkamp, N.; and Asfour, T. 2016. Heuristic 3d object shape completion based on symmetry and scene context. 74–81.

[24] Troje, N. F. 2002. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision* 2(5):2–2.