This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

4D Human Body Correspondences from Panoramic Depth Maps

Zhong Li*[1] Minye Wu*[2] Wangyiteng Zhou[2] Jingyi Yu[1,2]

[1]University of Delaware, Newark, DE, USA. lizhong@udel.edu

[2]ShanghaiTech University, Shanghai, China. {wumy,yujingyi}@shanghaitech.edu.cn
wytzhou@gmail.com

Abstract

The availability of affordable 3D full body reconstruction systems has given rise to free-viewpoint video (FVV) of human shapes. Most existing solutions produce temporally uncorrelated point clouds or meshes with unknown point/vertex correspondences. Individually compressing each frame is ineffective and still yields to ultra-large data sizes. We present an end-to-end deep learning scheme to establish dense shape correspondences and subsequently compress the data. Our approach uses sparse set of "panoramic" depth maps or PDMs, each emulating an inward-viewing concentric mosaics (CM) [45]. We then develop a learning-based technique to learn pixel-wise feature descriptors on PDMs. The results are fed into an autoencoder-based network for compression. Comprehensive experiments demonstrate our solution is robust and effective on both public and our newly captured datasets.

1. Introduction

There is an emerging trend on producing free-viewpoint video (FVV) of dynamic 3D human models [12], to provide viewers an unprecedented immersive viewing experience. The technology is largely enabled by the availability of affordable 3D acquisition systems and reliable reconstruction algorithms. The earlier attempt of Kanade et al [21] mounted 51 cameras on a 5 meter diameter dome to "virtualize" reality. More recent solutions can be viewed as it variations but using higher resolution, higher speed industrial cameras and easy-to-use synchronization schemes. For example, the CMU Panoptic studio [20] uses 480 cameras and can recover interactions between multiple human subjects. Active solutions such Microsoft Holoportation [34] further employ structured light to reduce the number of cameras.

*These authors contribute to the work equally.

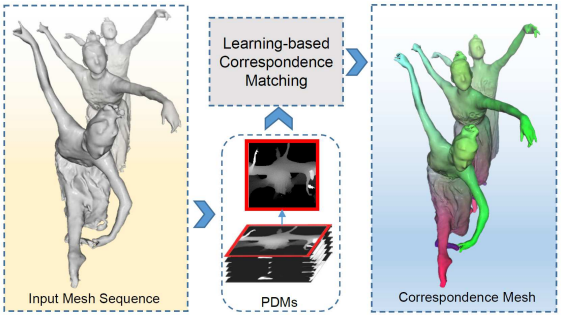


Figure 1: Our human body correspondence technique first renders Panoramic Depth Maps (PDMs) of the input mesh sequences and then conduct learning-based correspondence matching on the PDMs.

Despite heterogeneity in the digitization processes, a common challenge in FVV is the size of the reconstructed 4D data: each frame corresponds to a dense 3d mesh and a high resolution texture map and a short clip can easily lead to gigabytes of data if not compressed. For example, a sample clip of 10 seconds released by 8i [17] is 2 gigabytes. The large data size prohibits real-time transfer to or even storage on user-end devices. Although existing video compression standards can compress the texture maps and potentially the mesh, they ignore geometric consistencies and yields to low compression rate and quality.

The key to geometry-consistent compression is reliably establishing correspondences between geometric shapes. On 4D human body (geometry + time), the task is particularly challenging as such scans exhibit high noise, large non-rigid deformations, and topology changes. Existing approaches assume small deformations so that sparse shape descriptors [40] can be adopted. In reality, sparse shape descriptors fail on noisy data. Alternative dense shape correspondence schemes can reliably handle noise but require zero-genus surfaces, i.e., they are inapplicable to topology

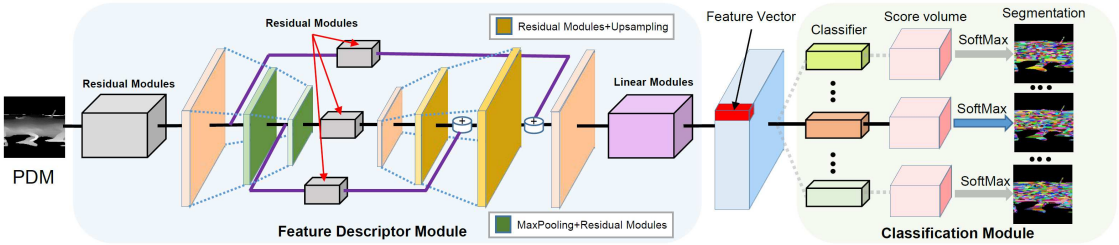


Figure 2: Our human body correspondences network structure extends the hourglass network. The feature descriptor module learns per-pixel feature vectors on the PDMs where the results, along with body segmentations, are fed into the classification module.

changes. A notable exception is the recent deep-learning based approach [50] that first trains a feature descriptor on depth maps produced from a large number of viewpoints 144 views to classify the body regions.

In a similar vein, we present an end-to-end deep learning scheme to conduct dense shape correspondences (as shown in Fig. 1) and subsequently compress the data. The key difference is that we aim to directly handle the complete 3D model without sampling depth maps from dense viewpoints. At each frame, we first produce a sparse set of "panoramic" depth maps or PDMs of 3D human model. Specifically, we construct 6 inward-viewing concentric mosaics (CM) [45] towards each model as shown in Fig. 3. Tradition CMs are synthesized by rendering an outward viewing cameras that lie on a common circle and then stitching the central column of each image into a panorama. The collected rays are multi-perspective and hence ensure the sampling of visible and occluded regions.

To conduct efficient training, we apply GPU-based multi-perspective rendering [54] of PDMs for a variety of 3D human sequences (the MIT dataset [49], SCAPE [4], and Yobi3D [50]). Next, we extend the hourglass networks [33] to learn a pixel-wise feature descriptor for distinguishing different body parts on PDMs. We further add a regularization term in the network loss function to maximize the distance between feature descriptors that belong to different body parts. Once we obtain the pixel-wise feature descriptor on each pixel of the PMDs, we back-project it onto the 3D models for computing vertex-wise feature descriptors.

Since each vertex can be potentially mapped to a pixel in each of the 6 PDMs, we set out to find the most consistent matching vertex pairs across all 6 PDMs via a voting scheme. To further remove the outliers, we conduct correspondence fine-tuning based on the observation that the trajectory of each vertex induced by matching should be temporally smooth. We also improve the accuracy of correspondences by enforcing the geodesic constraints [18]. The process can significantly reduce the outliers while maintaining smooth motion trajectories. Finally, we feed the correspondences to an autoencoder-based network for geometry compression.

We conduct comprehensive experiments on a wide range of existing and our own 3D human motion sequences. Compared with [50], the use of the PDMs significantly reduces the training data sizes. More importantly, PDMs are able to robustly handle occlusions in body geometry, yielding to a more reliable feature descriptor for correspondence matching. On the FAUST benchmark [5] our technique outperforms the state-of-the-art techniques and at the same time avoids complex optimizations. Further, our neural network based compression/decompression scheme achieves very high compression rate with low loss on both public and our newly captured datasets.

2. Related Work

The key to any successful dynamic mesh compression scheme is to establish accurate vertex correspondences. Lipman et al. [27] and Kim et al. [24] employed conformal geometric constraint between two frames. Such techniques are computationally expensive and require topological consistency. Bronstein et al. conduct vertex correspondence matching by imposing geodesic [9] or diffusion [8] distance constraints. Their techniques, however, still assume that input surfaces are nearly isometric and therefore cannot handle complex, articulated human motions. More recent approaches aim to design feature descriptors [51, 28] matching points. Pottemann et al. [36] use local geometric descriptors that can handle small motions. Taylor et al. [46] use random decision forest based approaches to infer correspondences.

Bogo et al. [5, 6] build a high-quality inter-shape correspondence benchmark by painting the human subject with high-frequency textures. Chen et al. [11] manage to use the Markov Random Field (MRF) to solve correspondence matching analogous to stereo matching. Yet their technique is vulnerable to human wearing cloths. Most recently learning-based approaches such as the ones based on anisotropic convolutional neural network [7] have shown

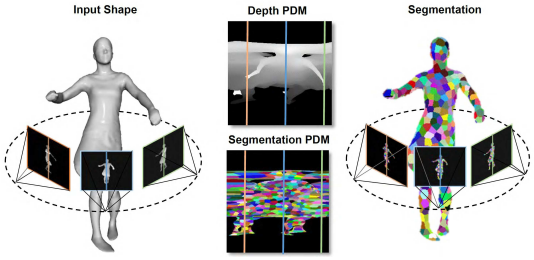


Figure 3: A PDM represents an omni-directional depth map of a 3D mesh. We can also generate the corresponding PDM segmentation map (color coded).

promising results on mesh correspondence matching. Yet the state-of-the-art solution [50] requires sampling the mesh from a large number of viewpoints (144 in their solution) to reliably learn per-pixel feature descriptor. In contrast, we show how to train a network using a few of panoramic images. Further, the focus of their work is on dense correspondence rather than compression as ours where we show the latter requires higher temporal coherence.

Different from mesh correspondence matching, animated mesh compression is a well-studied problem in computer graphics. State-of-the-art solutions however assume consecutive meshes have exact connectivity [30]. [15, 31] conduct pre-segmentation on the mesh to ensure connectivity consistencies. PCA-based methods [2, 47, 44, 29] aim to identify different geometric clusters of human body (arms, hands, legs, torso, head, etc). Spatio-temporal analysis [1, 19, 29] predict vertex trajectories for forming vertex groups. There is by far only a handful of works [16, 52] focusing directly on compressing uncorrelated mesh sequences, i.e., meshes without correspondences. The quality of these approach fall short compared with the ones with correspondences. Our learning based approach in contrast can robustly and accurately process uncorrelated mesh sequences and is able to achieve very high compression rate. Further, we employ the smoothness of the correspondence trajectory as well as geodesic consistencies for fine tuning our solution.

3. Vertex Correspondence Matching

We first present a learning-based scheme to establish vertex correspondences between a pair of models. Since pose changes can greatly affect the appearance of the model, previous approaches train on a very densely sampled viewpoints in the hope that some viewpoints will match the observed ones. We instead introduce a new panoramic depth map (PDM) for efficient training.

3.1. PDM Generation

A PDM, in essence, is a panoramic depth map of an inward-looking concentric mosaics (CM) [45], as shown in figure 3. Our key observation is that each CM covers the complete longitudinal views towards an object, capturing its omni-directional appearance. Traditionally, a CM can be synthesized by first rendering a dense sequence of images on a circle facing towards the human shape and then composing columns at identical locations (e.g., the middle column) from all images. Such a rendering is computationally expensive as it requires rendering a very large number of images.

We instead adopt a GPU-based multi-perspective rendering technique [54]: on the vertex shader, we map each vertex of the mesh onto the pixel using the CM projection model while by passing the traditional perspective projection. We also record the depth of the vertex by computing the distance between the vertex and the cylindrical image plane. The rasterization pipeline then automatically generates the PDM via interpolation, and with z-buffer enabled, we resolve the visibility issue when multiple triangles cover the same pixel. Our GPU algorithm is significantly more efficient than the composing approach. In our implementation, we render 6 PDMs at different latitudes, all viewing towards the center of the human object, at 20,30,40,50 and 60 degrees respectively. We find they are sufficiently robust for handling complex occlusions across a variety of poses.

Compared with prior art [50] that renders dense perspective depth maps, we use a much smaller images (6 vs. 144) where each PDM provides an omni-directional view towards the human object. More importantly, the PDM better handles occlusions than a regular perspective image. For example, in perspective depth maps, visible shape parts such as the head and the outer surface of the arms and legs appear many more times than "hidden" parts such as the inner parts of the arms and legs, causing not only redundancies but also bias on training. In contrast, in each PDM, each vertex will appear at most once, providing a more reliable training set for extracting shape descriptors.

3.2. Correspondence Matching

Next, we train a deep network on the PDMs for computing dense vertex correspondence over temporal frames of the human shape. We formulate the problem as follows: given two sets of vertices on the reference and the target meshes $\mathcal{S} = \{s|s \in \mathbb{R}^3\}, \mathcal{T} = \{\tau|\tau \in \mathbb{R}^3\}$, we set out to find a dense mapping between vertices. Clearly, this mapping should be able to differentiate body parts in order to establish reliable correspondences. We formulate the correspondence matching problem as a classification task: our goal is to train a feature descriptor $f : I_{PDM} \rightarrow \mathbb{R}^d$, which maps each pixel in every input PDM to a feature vector. For a pair of vertices that belong to the same or nearby anatomi-

cal parts, the difference between their feature vectors should be relatively small. For the ones that different anatomical parts, the difference should be large, especially for parts that lie far away from each other.

We construct a network with two modules: the feature descriptor module and the classification module.Fig. 2 shows our network architecture. We indirectly train f with the help of the classification module. To enforce smoothness of the feature descriptor, we partition the classification module into multiple segmentation tasks, one classifier per segmentation. Each classifier aims to assign every vertex and its correspondences across the mesh sequences with an identical label.

We train the feature descriptor f and classifiers simultaneously by adopting the loss function $L_{total} = L_{data} + L_{reg}$, where the data term L_{data} aims to resolves classification as

$$\boldsymbol{L_{data}} = -\frac{1}{\mathfrak{N}} \sum_{i}^{\mathfrak{N}} \sum_{p \in I_i} \sum_{j}^{K} [\mathbb{1}\{I_{i,p}^{seg} == j\} log(\phi_j^{m(i)}(f(p)))] \tag{1}$$

where

$$\phi_j^{m(i)}(x) = \frac{\exp(\theta_{m(i),j}^T x)}{\sum_l^K \exp(\theta_{m(i),l}^T x)} \tag{2}$$

$\mathfrak{N}$ corresponds to training batch size, i refers to the index of a training sample within a batch, I_i corresponds to the i_{th} input PDM image in current training batch, $I_{i,p}^{seg}$ is the label of pixel p in the i_{th} sample of training batch, K is the number of labels and M is the number of segmentations. $\theta_{m(i),l}$ refers to the parameters of classifier for segmentation $m(i)$. Eq.2 can be viewed as an extended Softmax regression model. $\mathbb{1}\{\cdot\}$ is the indicator function, so that $\mathbb{1}\{\text{a true statement}\} = 1$, and $\mathbb{1}\{\text{a false statement}\} = 0$.

The regularization term L_{reg} aims to make the feature descriptor more distinctive over different anatomical parts as:

$$\boldsymbol{L_{reg}} = -\sum_{i}^{\mathfrak{N}} \sum_{\mu} \sum_{\nu > \mu} ||avg(f(I_i^{l_\mu})) - avg(f(I_i^{l_\nu}))||^2 \tag{3}$$

where l_μ and l_ν are the μ_{th} and ν_{th} label's mask, respectively. $f(I_i^{l_\mu})$ and $f(I_i^{l_\nu})$ are the feature vectors labeled with μ and ν. $avg(\cdot)$ calculates the average over a sets of vectors.

While several recent approaches have adopted the Alexnet [2] as the feature descriptor module, we recognize that our problem resembles more the human pose estimation problem where hourglass [33] has shown superb performance. Its network structure is capable of downsampling feature maps at multiple scales where it can process each scale via convolutional and batch normalization layers. In our implementation, we conduct nearest neighbor upsampling on feature maps to match across different scales. After each upsampling, a residual connection transmits the information from the current scale (level) to the upsampled scale via element-wise addition. Finally, feature vectors can be extracted across the scales in a single pipeline with the skip layers. We also remove the first max pooling layer to match output resolution with the input.

Algorithm 1 Improve Correspondence Matching Via Voting

procedure $Voting(\mathcal{S}, \mathcal{T}, f_s, f_t, I_s, I_t)$
  Initial vote matrix $m_{vote} = zeros(len(f_s^p), len(f_t^q))$
  for `each source view` p **do**
    for `each target view` q **do**
      P_s^p = *Reproject*(I_s^p)
      P_t^q = *Reproject*(I_t^q)
      $index_s^p$ =*nnsearch*($\mathcal{S}$,P_s^p)
      $index_t^q$ =*nnsearch*($\mathcal{T}$,P_t^q)
      $\mathcal{F}_{index}$ =*nnsearch*(f_t^q,f_s^p)
      for $k = 1 : len(f_s^p)$ **do**
        $vote_s = index_s^p(k)$
        $vote_t = index_t^q(\mathcal{F}_{index}(k))$
        $m_{vote}(vote_s, vote_t)$ += 1
  for `each row in` m_{vote} **do**
    CorresIdx(row) = $argmax_j(m_{vote}(row, j))$
  return $CorresIdx$

Recall that previous methods average the per-pixel feature vectors to obtain a per-vertex feature vector. We adopt a different scheme: assume shape $\mathcal{S}$ and $\mathcal{T}$ have sets of feature vectors f_s and f_t respectively, where f_s^p and f_t^q correspond to feature vectors in shape $\mathcal{S}$,$\mathcal{T}$, and the view p,q of the 3D model; we build a $len(f_s^p) \times len(f_t^q)$ voting matrix that matches correspondences from the source view p's feature vectors f_s^p to the target view q's feature vectors f_t^q via nearest neighbor search in feature space. We then accumulate the votes into a voting matrix. Finally, we extract the maximum vote index of each row as final correspondence $CorresIdx$. An outline of the algorithm is shown in Algorithm 1, where $Reproject$ projects PDM I_s and I_t onto 3D points P_s^p and P_t^q, $index = nnsearch(X, Y)$ finds the nearest neighbor in X for each point in Y, each row in $index$ corresponds to the index of nearest neighbor in X of the corresponding row in Y.

3.3. Implementation

For training, we collect training data from the MIT dataset [48], SCAPE [4] and Yobi3D [50]. The MIT dataset contains 10 human objects with the ground truth dense correspondences and we use 7 out of 10 for training (samba,march1,squat1,squat2,bouncing,crane,march1) and the rest 3 (swing, jumping and march2) for testing. SCAPE models a single human subject of 71 different poses where the poses are registered to form dense correspondences. Yobi3D [50] consists of 2,000 avatars in varies poses. To generate segmentation patches, for the MIT dataset and

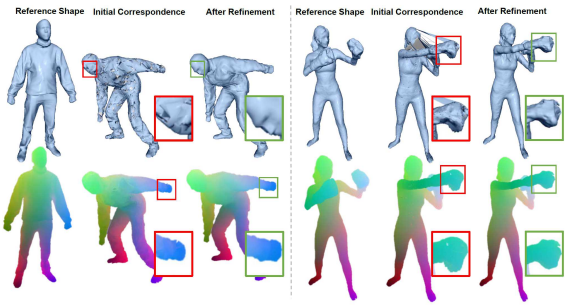


Figure 4: Result before and after refinements in the Jumping and Boxing sequences. For each sequence, from left to right we show the reference mesh, the target mesh with initial correspondence matching, the target mesh after refinement. The top and bottom rows show the geometry and correspondence maps respectively.

SCAPE we follow the same strategy as [50] by segmenting each model into 500 patches. For each mesh sequence with the ground truth correspondences, we generate each segmentation by randomly select 10 points on each model. We then add the remaining points using farthest point-sampling and obtain the segmentation by using those sample points as cluster center. Finally, we propagate the initial segmentations onto consecutive frames using the known dense correspondences. For yobi3D data, recall that no dense correspondences are available across the models, we use manually semantic annotated key points as the cluster centers to generate segmentations.

Our feature extraction module takes the PDMs as input and is trained on 2-cascaded-level hourglass network with the first max-pooling layer removed. In each classification layer, we use a convolution layer with 1×1 filter size to replace fully-connected layer and conduct 2D Softmax operation on feature vectors generated by descriptor. Different classifier is trained for different body parts across all frames in the mesh sequence but the descriptor's parameters remain the same. Our network is rather large on high resolution PDMs(512×512 resolution) and we handle this by training a batch size of 4. The training time is approximately 48 hours on a single Titan X Pascal GPU.

4. Correspondence Refinement

Our correspondence matching scheme processes any two consecutive frames of meshes. We further refine the results to handle a N-frame motion sequence $\{P_n\}_{n=1}^{N}$ where dense correspondences are maintained coherently across the frames. The challenges in processing real data is that they contain noise and topological inconsistencies. We therefore first locate a reference frame P_κ that has the lowest genus. Assume the mesh contains V vertices, to reduce drifting we compute correspondences $\{c_i^n\}_{i=1}^{V}$ between P_k and every other frame P_n using our feature descriptor. As a result, we obtain a vertex trajectory matrix A that stores each frame in its columns:

$$\boldsymbol{A} = \begin{pmatrix} c_1^1 & c_1^2 & \dots & c_1^N \\ c_2^1 & c_2^2 & \dots & c_2^N \\ \dots & \dots & \dots & \dots \\ c_V^1 & c_V^2 & \dots & c_V^N \end{pmatrix} = \begin{pmatrix} T_1 \\ T_2 \\ \dots \\ T_V \end{pmatrix} \tag{4}$$

where T_i is a row vector that represents each vertex correspondence trajectory $\{c_i^n\}_{n=1}^{N}$.

Figure 4 illustrates that, even majority of the correspondences are accurate, a small percentage of outliers can cause severe distortions on mesh surfaces. We therefore conduct a correspondence refinement step on the vertex matrix A using geodesic and temporal constraints similar to [18]. By assuming the deformation is isometric, we first find the correspondence outliers c_i^n in each vertex trajectory T_i based on the geodesic distance consistency measurement. We then refine each outliers by imposing temporal smoothness $E_{temporal}$ and geodesic consistency $E_{geodesic}$ constraints as:

$$\boldsymbol{E_{temporal}} = ||c_i^{(n+1)} + c_i^{(n-1)} - 2t||^2 \tag{5}$$

and

$$\boldsymbol{E_{geodesic}} = \sum_{(c_k^n, c_k^{n\pm1}) \in \Theta} [d_g(c_i^{n-1}, c_k^{n-1}) - d_g(t, c_k^n)]^2 + [d_g(t, c_k^n) - d_g(c_i^{n+1}, c_k^{n+1})]^2 \tag{6}$$

where d_g is the geodesic distance between two points, Θ is the set of confident correspondence set among three frames $n-1, n, n+1$. To enforce geodesic consistency, given a trajectory outlier c_i^n in vertex trajectory i, we find the nearest c_j^n that has a highly confident correspondence to c_j^{n-1} and c_j^{n+1}, and assign c_j^n as the adjusted position t to replace c_i^n. Next, we construct the geodesic term $E_{geodesic}$ to enforce the geodesic distance between each pair of correspondence in frame n close to each pair of correspondences in frame $n-1$ and $n+1$. To enforce temporal smoothness, we utilize a temporal term $E_{temporal}$ by assuming each the outlier vertex c_i^n in frame n should close to the middle of the two adjacent corresponding vertex c_i^{n+1} and c_i^{n-1} respectively. To find an optimal t to minimize $E_{refine} = E_{temporal} + E_{geodesic}$, we set out to refine each individual correspondence outlier c_i^n by searching the $\mathcal{K}$ nearest neighbor of $t \in \Omega(c_j^n)$. Figure 4 compares the results before and after the refinement: the reconstructed mesh surface contains much fewer artifacts after refinement.

5. Mesh Compression and Decompression

With the refined vertex correspondences, we can effectively convert the input mesh sequence to an animation mesh with a consistent topology. Recall that an animation sequence should have consistent vertices and con-

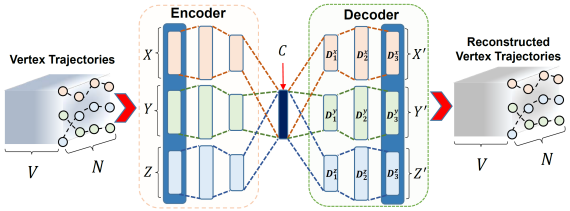


Figure 5: Our Autoencoder network for mesh compression/decompression. The x, y, and z dimensions of the vertex trajectories are separately processed. Each block, except the output one, represents a fully-connected layer with ReLU as its active function.

nectivities, we simply need to compress vertex trajectories $A = \{T_i\}_{i=1}^{V}$.

Traditional dimension reduction techniques are commonly used for compressing animation meshes. We adopt the Autoencoder framework, as shown in Figure 5. we present a 7-layer parallel Autoencoder network structure with 3D vertex trajectories as input. In the encoder path, to encode the trajectory according to x, y, z coordinate separately, we split $\{T_i\}_{i=1}^{V}$ into three parts: $\{T_i^x\}_{i=1}^{V}$,$\{T_i^y\}_{i=1}^{V}$ and $\{T_i^z\}_{i=1}^{V}$, and feed the above three parts into three parallel networks respectively. The three parallel network will then merge into an intermediate layer C which is a compressed representation of the input data.

The decoder path is the inverse operation of encoder. For decompression, we efficiently extract the trained parameters from the intermediate layer C and the rest of decoder layer $D_1^{x,y,z}$, $D_2^{x,y,z}$, $D_3^{x,y,z}$ to conduct a forward operation to reconstruct the entire animation sequence. In our training process, we construct layers with varying sizes to achieve different bpvf (bit per frame per vertex). Our training process uses a batch size of 200, and the process converges after about 6,000 iterations on the GPU. Compared with traditional Principle Component Analysis (PCA) based approaches, our solution supports nonlinear trajectories and therefore is much more effective in both compression rate and quality, as shown later in the experiment.

6. Experimental Results

We conduct comprehensive experiments on both publicly available datasets [49, 5] and our own 4D human body dataset. Our own capture system is composed of 32 synchronized industry cameras at a 720P resolution and 24 fps. We then apply structure-from-motion for camera pose calibration and subsequently use open-source multi-view stereo matching solutions [13] for generating the point cloud. To construct the mesh from the point cloud, we apply Poisson surface reconstruction [23]. Notice that the initially reconstructed meshes do not have vertex correspondences. In the paper, we demonstrate 3 full

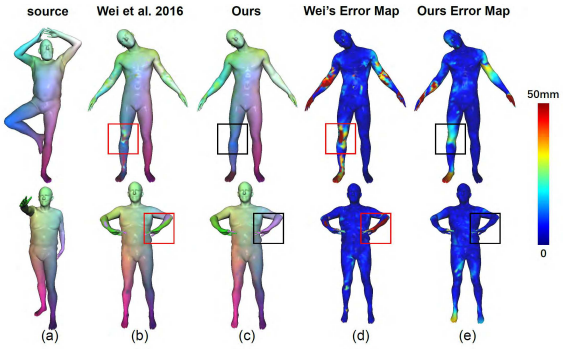


Figure 6: Comparisons on FAUST [5]. (a) shows the reference mesh; (b) and (c) show the results by [50] and ours; (d) and (e) show the corresponding errors maps. Our technique more robustly handles strong deformations (e.g., knee and elbow bending).

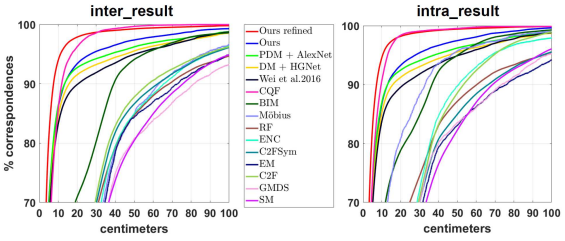


Figure 7: Quantitative evaluations on FAUST [5]. Our technique outperforms the state-of-the-art [50], especially after the refinement process.

body motion sequences:Yoga,Wakingup and Boxing has 360,300,270 frame respectively, each data with average 32K vertex number per mesh.

Our pipeline conducts the three-step approach to simultaneously compute the dense correspondences across the frames and then use the results to compress the mesh sequence. All experiments (including training and testing) are performed off-line on a PC with CPU Intel Core i7-5820K, 32 GB memory and a Titan X GPU.On the computation overhead, the average cost for mesh correspondence generation process is 21 secs for establishing the initial dense correspondences and 11 secs for correspondence refinement. Our mesh compression step takes about average 64 secs for compressing each entire sequence and 5 secs for decompression.

Correspondence Matching Results. To further demonstrate the effectiveness of our learning-based correspondence matching technique, we experiment on the FAUST dataset [5]. FAUST is a public dataset composed of training and testing data. The training data has ground truth dense correspondences across the frames but the testing data does not. We conduct the experiment on the training

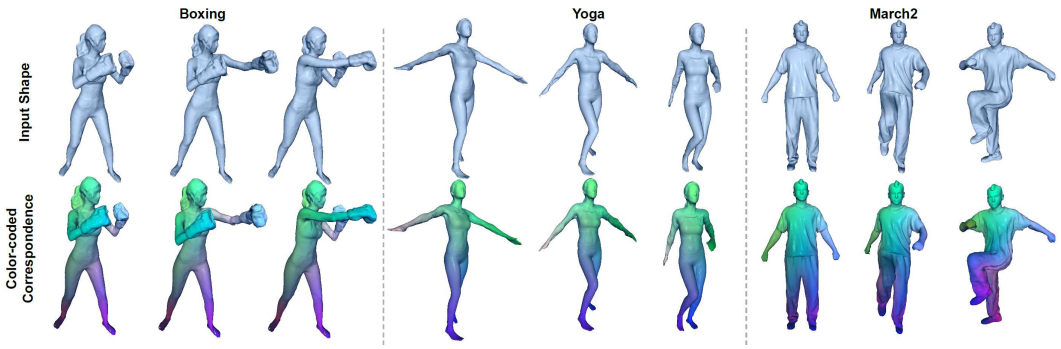


Figure 8: Results using our technique on the Boxing, Yoga datasets and MIT [49] March2 datasets.

dataset that has 100 shapes, includes 10 human subjects with 10 different poses. We have conducted two different types of evaluation. The first computes the correspondence between inter-object: source and target are of different human subjects with the different poses and the second between intra-object: source and target are of the same human object but with the different poses. The results shown in Fig. 6 demonstrate our method incurs less error compared to the state-of-the-art [50] based on our own implementation. In Fig. 7, we conduct quantitative evaluations and show error distributions in centimeters. Other techniques including GMDS [8], Mobius voting [27], blended intrinsic maps (BIM) [25], coarse-to-fine matching (C2F) [43], the EM algorithm [41], coarse-to-fine matching with symmetric flips (C2FSym) [42], sparse modeling (SM) [35], elastic net constraints (ENC) [39], and random forests (RF) [38] were based on the implementations by Chen et al. [11]. Figure. 7 shows comparisons on accuracy of our technique vs. others on all intra-subject pairs and all inter-subject pairs. We also conduct self-evaluation to compare the modified hourglass architecture vs. [50] with the same PDM inputs. As shown in Figure. 7, the hourglass architecture outperforms [50] using either traditional depth maps or PDMs as inputs while PDMs still significantly outperform the regular depth maps.

Next we show how our refinement step further improves the correspondence matching results in Fig. 9. We only evaluate this process on the MIT dataset, the only one with ground truth dense correspondences. The first column in Fig. 9 shows the reference and target meshes from up to bottom. Starting from the second column, we first show the results [14] among the state-of-the-art non-rigid surface alignment methods [32, 3, 10, 26], previous CNN approach [50], our technique before refinement, and ours after refinements respectively; the second row shows the corresponding error maps using various techniques. We observe that due to large deformations between frames, [14] can lose tracking and produce relatively large errors. Our technique before refinement already contains fewer artifacts compared

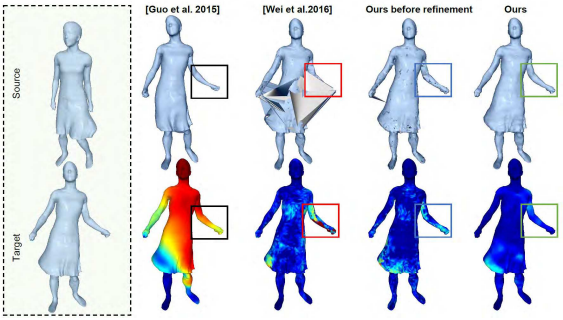


Figure 9: Our technique vs. the state-of-the-art on the Swing sequence. [14] loses track due to large deformations of the skirt (see geometry inconsistencies); [50] is able to track most of the vertices but the errors produce topology inconsistencies; Our result before refinement outperforms both in correspondence matching. The results are further improved after refinements. The video results can be found in the supplementary materials.

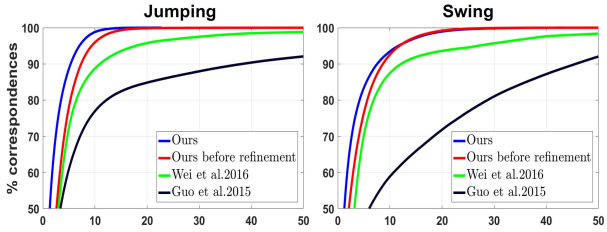


Figure 10: Quantitative Comparisons of ours vs. [14] and [50] on highly non-rigid Jumping and Swing sequences.

with the previous learning-based approach [50]. The artifacts were further reduced after refinement. Fig.10 shows the quantitative evaluations using our method vs. prior art on the jumping and swing sequences. Fig. 8 shows additional results on Boxing, Yoga and March2.

Mesh Compression Results. Based on the vertex correspondences, we construct animation mesh sequences with

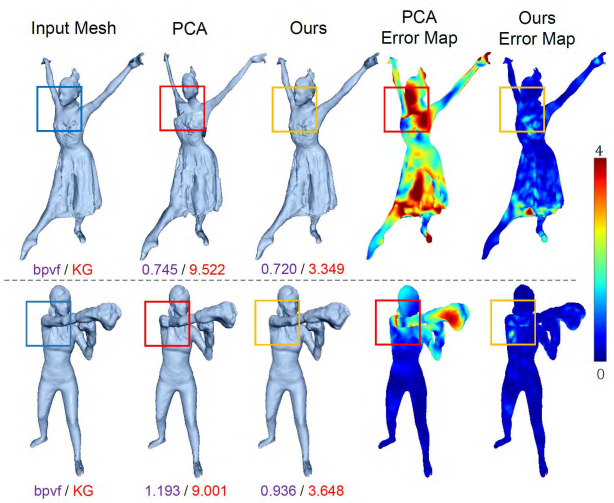


Figure 11: Visual comparisons of ours vs. the PCA technique [44] on mesh compression. Our technique outperforms [44] even with at lower bpvf.

consistent connectivity and apply our Autoencoder neural network efficient data compression. We conduct experiments on six motion sequences (jumping, march2, swing, yoga, boxing, and dance). We first the evaluate the quality degradation caused by compression. Specifically, we measure the distortion errors between the original and the (de)compressed results by using the well-established vertex-based error metric: KG error [22]. KG error measures the quality of vertex position reconstruction over an entire sequence as: $\mathbf{D_{KG}} = 100 \cdot \frac{\|B - \hat{B}\|}{\|B - E(B)\|}$ where B is a matrix representing the original mesh sequence data with size $3V \times N$, $\hat{B}$ is same sized matrix that stands for the mesh sequence after reconstruction. $E(B)$ is a matrix consist of the average vertex position for all frames. Fig.11 shows the comparisons on our technique vs. Sattler et al. [44] (our own implementation) at different $bpvf$ (bits per vertex per frame) , we will shown the our bpvf metric in supplementtal material. Fig. 11 and Fig. 12 shows that [44] introduces relative high distortions and errors when $bpvf$ is low whereas our Autoencoder approach significantly suppresses the errors.

We further measure the error between the original input mesh sequences and mesh sequences after decompression. It is important to note that there is no longer dense correspondences after we decode the compressed mesh sequences. We therefore use the mean Hausdorff distance $\mathcal{H}$ to measure the geometric deviations between the original and the decoded meshes as $\mathbf{D_{Hausdorff}} = \sum_i^N \mathcal{H}_i$, where N is the frame number of the sequence, $\mathbf{D_{Hausdorff}}$ computes the average Hausdorff distance of entire sequences at different $bpvf$. Fig. 12 shows that our decoded mesh sequence has a very low Hausdorff error for over various motion sequences.

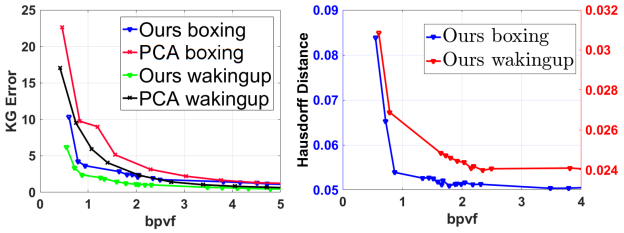


Figure 12: Quantitative comparisons of mesh compression on the Boxing and Wakingup sequences using KG error and Hausdorff distance error measures.

7. Conclusions and Future Work

We have presented a learning-based approach for compressing 4D human body sequences. At the core of our technique is a novel temporal vertex correspondence matching scheme based on the new representation of panoramic depth maps or PDM. The idea of PDM is borrowed from earlier panoramic rendering techniques such as concentric mosaics [45] and multi-perspective rendering [37, 53] that samples omni-directionally the appearance of a target object (in our case the depth map of an human body). By extending existing deep learning frameworks, our technique manages to learn how to reliably label vertices into meaningful semantic groups and subsequently establishes correspondences. We have further developed an autoencoder-based network that directly uses correspondences for simultaneous texture and geometry compression. Regarding the limitation, topology changes and occlusions may cause the correspondence tracking failure. A potential solution is to partition the sequence into shorter, topologically coherent segments.

Alternatively FVVs can be produced via image-based rendering such as view morphing where new views can be synthesized by interpolating from acquired reference views without completely obtaining the 3D geometry. Our immediate future task hence is to extend our approach to handle such cases. We also plan to experiment on applying our technique for 3D completion: a partial scan, e.g., a depth map of the model, can be registered onto a reference, complete model using our technique and missing parts can be completed via warping.

Acknowledgement

This work is partially supported by National Science Fundation under the Grant CNS-1513031. Majority of the work was performed while Zhong Li was an intern at Plex-VR Inc.

References

[1] J.-K. Ahn, Y. J. Koh, and C.-S. Kim. Efficient fine-granular scalable coding of 3d mesh sequences. *IEEE Trans. Multimedia*, 15:485–497, 2013. 3

[2] M. Alexa and y. v. p. Wolfgang Müller, journal=Comput. Graph. Forum. Representing animations by principal components. 3, 4

[3] B. Allain, J.-S. Franco, and E. Boyer. An efficient volumetric framework for shape tracking. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 268–276, 2015. 7

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24:408–416, 2005. 2, 4

[5] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 2, 6

[6] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic faust: Registering human bodies in motion. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5573–5582, 2017. 2

[7] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016. 2

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006. 2, 7

[9] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2):266–286, 2010. 2

[10] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global non-rigid alignment of surface sequences. *International Journal of Computer Vision*, 102:256–270, 2012. 7

[11] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2039–2047, 2015. 2, 7

[12] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. G. Kirk, and S. J. Sullivan. High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34:69:1–69:13, 2015. 1

[13] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 6

[14] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3083–3091, 2015. 7

[15] S. Gupta, K. Sengupta, and A. A. Kassim. Compression of dynamic 3d geometry data using iterative closest point algorithm. *Computer Vision and Image Understanding*, 87(1-3):116–130, 2002. 3

[16] S.-R. Han, T. Yamasaki, and K. Aizawa. Time-varying mesh compression using an extended block matching algorithm. *IEEE Trans. Circuits Syst. Video Techn.*, 17:1506–1518, 2007. 3

[17] https://8i.com/. Real human holograms for augmented, virtual and mixed reality. *Accessed:2017-10-03*. 1

[18] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. In *Computer Graphics Forum*, volume 27, pages 1449–1457. Wiley Online Library, 2008. 2, 5

[19] L. Ibarria and J. Rossignac. Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity. In *Symposium on Computer Animation*, 2003. 3

[20] H. Joo, H. Liu, L. Tan, L. Gui, B. C. Nabbe, I. A. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3334–3342, 2015. 1

[21] T. Kanade and P. J. Narayanan. Virtualized reality: Perspectives on 4d digitization of dynamic events. *IEEE Computer Graphics and Applications*, 27, 2007. 1

[22] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *EuroCG*, 2000. 8

[23] M. M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 2006. 6

[24] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser. Möbius transformations for global intrinsic symmetry analysis. In *Computer Graphics Forum*, volume 29, pages 1689–1700. Wiley Online Library, 2010. 2

[25] V. G. Kim, Y. Lipman, and T. A. Funkhouser. Blended intrinsic maps. *ACM Trans. Graph.*, 30:79:1–79:12, 2011. 7

[26] Z. Li, Y. Ji, W. Yang, J. Ye, and J. Yu. Robust 3d human motion reconstruction via dynamic template construction. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017. 7

[27] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. In *ACM Transactions on Graphics (TOG)*, volume 28, page 72. ACM, 2009. 2, 7

[28] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2014. 2

[29] G. Luo, F. Cordier, and H. Seo. Compression of 3d mesh sequences by temporal segmentation. *Journal of Visualization and Computer Animation*, 24:365–375, 2013. 3

[30] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot. 3d mesh compression: Survey, comparisons, and emerging trends. *ACM Computing Surveys (CSUR)*, 47(3):44, 2015. 3

[31] K. Mamou, T. Zaharia, and F. Prêteux. Tfan: A low complexity 3d mesh compression algorithm. *Computer Animation and Virtual Worlds*, 20(2-3):343–354, 2009. 3

[32] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352, 2015. 7

[33] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 2, 4

[34] S. Orts, C. Rhemann, S. R. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. T. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. P. C. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi. Holoportation: Virtual 3d teleportation in real-time. In *UIST*, 2016. 1

[35] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. Sparse modeling of intrinsic correspondences. *Comput. Graph. Forum*, 32:459–468, 2013. 7

[36] H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26(1):37–60, 2009. 2

[37] P. Rademacher and G. Bishop. Multiple-center-of-projection images. pages 199–206, 1998. 8

[38] E. Rodolà, S. R. Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4184, 2014. 7

[39] E. Rodolà, A. Torsello, T. Harada, Y. Kuniyoshi, and D. Cremers. Elastic net constraints for shape matching. *2013 IEEE International Conference on Computer Vision*, pages 1169–1176, 2013. 7

[40] S. Rusinkiewicz and B. J. Brown. 3d scan matching and registration. 2005. 1

[41] Y. Sahillioğlu and Y. Yemez. Minimum-distortion isometric shape correspondence using em algorithm. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2203–2215, 2012. 7

[42] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine isometric shape correspondence by tracking symmetric flips. In *Computer Graphics Forum*, volume 32, pages 177–189. Wiley Online Library, 2013. 7

[43] Y. Sahilliolu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1461–1470. Wiley Online Library, 2011. 7

[44] M. Sattler, R. Sarlette, and R. Klein. Simple and efficient compression of animation sequences. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217. ACM, 2005. 3, 8

[45] H. Shum and L. wei He. Rendering with concentric mosaics. In *SIGGRAPH*, 1999. 1, 2, 3, 8

[46] J. Taylor, J. Shotton, T. Sharp, and A. W. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–110, 2012. 2

[47] L. Vasa and V. Skala. Coddyac: Connectivity driven dynamic mesh compression. In *3dtv Conference*, pages 1–4, 2007. 3

[48] D. Vlasic, I. Baran, W. Matusik, and J. Popovic. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27:97:1–97:9, 2008. 4

[49] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics (TOG)*, 28(5):174, 2009. 2, 6, 7

[50] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016. 2, 3, 4, 5, 6, 7

[51] T. Windheuser, M. Vestner, E. Rodolà, R. Triebel, and D. Cremers. Optimal intrinsic descriptors for non-rigid shape analysis. In *BMVC*, 2014. 2

[52] T. Yamasaki and K. Aizawa. Patch-based compression for time-varying meshes. *2010 IEEE International Conference on Image Processing*, pages 3433–3436, 2010. 3

[53] J. Yu and L. Mcmillan. A framework for multiperspective rendering. In *Fifteenth Eurographics Conference on Rendering Techniques*, pages 61–68, 2004. 8

[54] X. Yu, J. Yu, and L. McMillan. Towards multi-perspective rasterization. *The Visual Computer*, 25:549–557, 2009. 2, 3