ViSER: Video-Specific Surface Embeddings for Articulated 3D Shape Reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce ViSER, a method for recovering articulated 3D shapes and dense 3D part trajectories from monocular videos. Previous work on high-quality reconstruction of dynamic 3D shapes typically relies on multiple camera views, strong category-specific priors, or 2D keypoint supervision. We show that none of these are required with reliable estimation of long-range 2D point correspondences, making use of only 2D object masks and two-frame optical flow as inputs. ViSER infers correspondences by matching 2D pixels to a canonical, deformable 3D mesh via *video-specific surface embeddings* that capture the pixel appearance of each surface point. These surface embeddings behave as a continous set of keypoint descriptors defined over the mesh surface, which can be used to establish dense long-range correspondences across pixels. These surface embeddings are implemented via surface-coordinate-based MLPs that are fit to each video via contrastive reconstruction losses. Experimental results show that ViSER compares favorably against prior work on challenging videos of humans with loose clothing and unusual poses as well as animals videos from DAVIS and YTVOS.

1 Introduction

Reconstructing the world from a sequence of monocular frames is a long-standing task in computer vision. While there has been tremendous progress in reconstructing rigid scenes (via SFM [36, 32], SLAM [7], or recent techniques based on neural rendering [24]), reconstructing *dynamic* scenes with articulated objects remains elusive. For example, given a monocular video, it is still challenging to reconstruct an everyday scene of a moving person with loose clothing. In this work, we tackle the problem of estimating the deforming mesh of articulated objects given a segmented monocular video of that object. Our method avoids the use of any mesh templates or category-specific priors and generalizes to unknown deformable articulated objects in the wild.

Nonrigid shape recovery is highly under-constrained due to fundamental ambiguities between shape, appearance, and time-varying deformation. Current approaches for addressing these challenges fall into two camps: better data "likelihoods" or better "priors". The first camp extracts richer sensor data, via multi-camera studio setups [14] or depth sensors [26], but requires substantial efforts to work in the wild. The second camp makes use of category-level priors over object shapes [17, 19] and is particularly effective for human reconstruction. However, building such models requires considerable offline efforts in the form of registered 3D scans [23] or manual keypoint annotations [11], both of which are difficult to scale to arbitrary object categories.

In this work, we use a practical but less explored variant of the first camp (that extracts better data likelihoods) where we use *multiple frames* of a video rather than multiple cameras or depth sensors. This considerably complicates analysis for dynamic, non-rigid scenes, spawning the field of nonrigid structure-from-motion (NRSfM) [4, 31], which typically relies on motion correspondences, such as

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

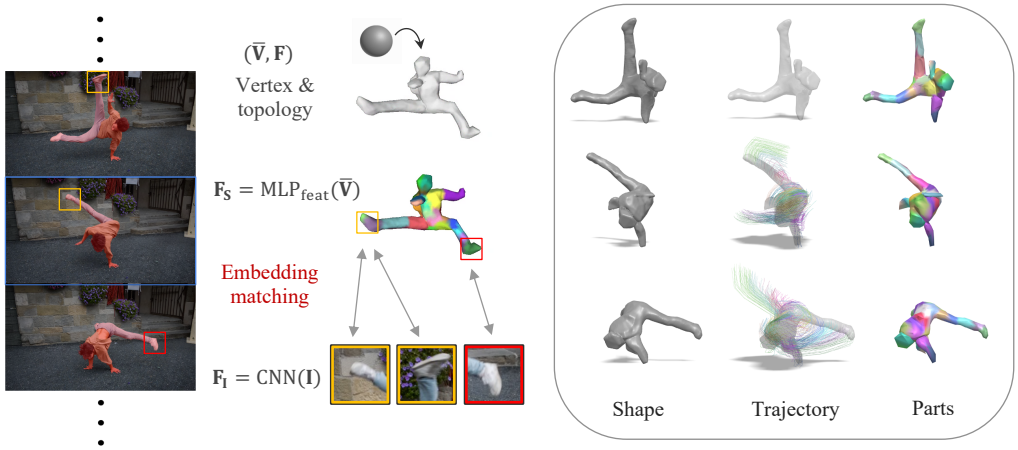

Figure 1: Given a long video or multiple short videos, we jointly learn articulated 3D shapes (represented as a mesh with vertices $\bar{\mathbf{V}}$ and faces $\mathbf{F}$) a joint pixel-surface embeddings (inlcuding a surface embedding $\mathbf{F_S}$ and a pixel embedding $\mathbf{F_I}$) to establish dense correspondences. As a result, accurate shape, long term trajectory and meaningful part segmentation can be recovered. Please see the supplementary material for the video results.

2D point tracks, to constrain the problem. While 2D correspondences over short time scales (i.e., optical flow) are relatively robust to extract, 2D correspondences over long time scales are notoriously difficult to estimate because of appearance variations arising from viewpoint changes, occlusion and fast motion. In practice, this limits the applicability of NRSfM methods to controlled lab sequences.

We propose ViSER (Video Surface Embeddings for Reconstruction), which establishes long-range correspondence and reconstructs articulated 3D shapes from a monocular video. Figure 1 shows a sample outdoor video and the corresponding ViSER results. The key insight behind ViSER is to force long-range video pixel correspondences to be consistent with an underlying canonical 3D mesh through the use of video-specific surface embeddings, as shown in Fig. 1. In order to scale to thousands of mesh vertices, each with its own embedding, ViSER infers a distributed representation with a low-dimensional embedding defined over the continuous mesh surface. These surface embeddings are implemented via surface-coordinate-based MLPs that are fitted to each video via contrastive reconstruction losses. ViSER simultaneously optimizes both 2D CNN and 3D MLP that computes the surface embeddings from video frames and inferred 3D shape respectively. ViSER reconstructs state-of-the-art articulated 3D shape and 3D trajectories without using category-specific priors, making it easily scalable to diverse videos including humans with loose clothing and animals. Lastly, we demonstrate that the inferred 3D trajectories can be used to recover detailed part segmentation and blend skinning weights, which typically require considerable manual effort from 3D artists.

2 Related Work

Dense video correspondence. Optical flow is a well-studied representation for short-term correspondence between adjacent frames of a video. After decades of research, recent CNN models [33, 40, 35] for optical flow have achieved an impressive level of accuracy as evidenced by the Sintel and KITTI benchmarks [5, 9]. However, it is challenging to concatenate optical flow for reliable long-range correspondence due to occlusions and strong appearance changes [29, 30, 34]. ViSER does not concatenate optical flow but use it as a constraint to establishes long-range correspondence.

The layered approach [6, 13, 38] segments a video into different moving objects with coherent motion, thereby establishing long-range correspondence for every frames through the shared layers. Early layered methods assume parameter motion for each layer and can only handle limited scenes. Unwrap Mosiacs [28] uses dense a 2D-to-2D mapping from a texture map to every input frame, and editing operations on the texture map naturally transfers to each individual frame. However, the 2D representation cannot flexibly model complex 3D phenomena, such as occlusions. By contrast, ViSER learns video surface embeddings to map every input frame to a canonical 3D shape and works on videos with occlusions and topology changes.

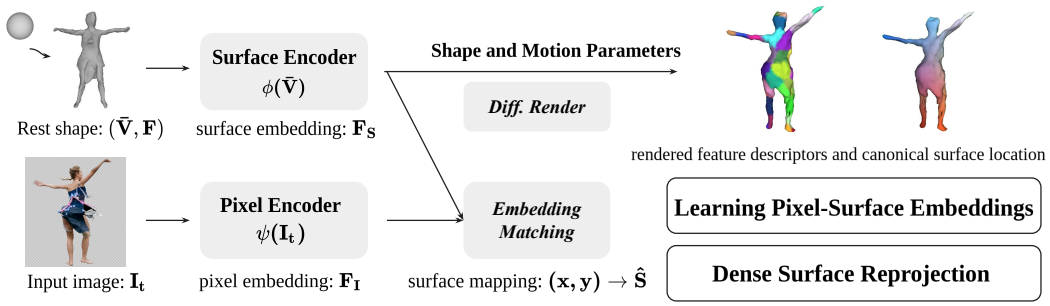

Figure 2: Given a test video, we learn joint pixel-surface embeddings for dense correspondence between pixels in video frames I_t and points on a canonical 3D surface $(\bar{\mathbf{V}}, \mathbf{F})$ (Sec. 3.2). Such embeddings are learned through differentiable rendering and correspondence matching, and enable articulated shape optimization through dense surface re-projection (Sec. 3.3).

Dense pose and surface mapping. DensePose [11] directly maps all human RGB pixels to the 3D surface of a human body. It requires large amounts of training data with annotated image-to-surface correspondence and is hard to generalize to other categories. Articulation-aware Canonical Surface Mapping (A-CSM) [19] uses geometric cycle consistency for learning to map pixels to corresponding points on a template shape without using keypoint annotations. However, it requires a pre-defined template shape for each category. Continuous Surface Embeddings (CSE) [25] establishes dense correspondences between image pixels and 3D object geometry by predicting an embedding vector of the corresponding vertex in the object mesh for each pixel in a 2D image. While applicable to multiple categories, CSE requires annotations and only applies to categories in the training set. ViSER requires neither a template shape nor annotations to work on categories in the wild.

Nonrigid shape reconstruction. One accurate way to reconstruct articulate shapes is to rely on rich sensor data, *e.g.*, multi-view [14] or depth sensors [26], which requires substantial efforts to work on videos in the wild. For monocular videos/images, one popular approach is to adopt strong priors [23, 17, 19] but it works on only limited categories. Template-free methods, such as nonrigid structure from motion (NRSfM) methods [4, 10, 18, 31] do not work well on videos in the wild due to the lack of reliable long-range correspondences [30, 34].

Recently LASR [41] uses optical flow to reconstruct articulate shapes from a monocular video. Despite the promising results, LASR does not reason about long-range correspondence and can only reconstruct what is visible in a short video. ViSER establishes reliable long-range correspondence that are robust to moderate shape variations and strong appearance changes. Thus, ViSER can obtain much higher-quality reconstruction by using either a long video or several videos of a category.

3 Approach

Figure 2 provides an overview of our approach, which follows the differentiable rendering framework [15, 22]. Borrowing the notation from LASR [41], we formalize our task as follows. Given a set of video observations including color, segmentation, and optical flow estimates $\{I_t, S_t, \mathbf{u}_t\}_{t=\{0,\ldots,T\}}$, our goal is to recover a set of shape and motion parameters $\{\mathbf{S}, \mathbf{D}_t\}$ that produce the corresponding renderings $\{\hat{I}_t, \hat{S}_t, \hat{\mathbf{u}}_t\}_{t=\{0,\ldots,T\}}$ to match the video observations.

3.1 Preliminaries

We represent an object's shape as a triangular mesh $\mathbf{S} = \{\bar{\mathbf{V}}, \mathbf{F}\}$ with canonical vertices $\bar{\mathbf{V}} \in \mathbb{R}^{N\times 3}$ and a fixed topology (edge connectivity) $\mathbf{F} \in \mathbb{R}^{M\times 3}$. To render an object, we displace mesh vertices with motion parameters $\mathbf{D}_t$, apply a perspective projection with camera intrinsics $\mathbf{K}_t$, and rasterize.

We model vertex motion with root body transformations $\mathbf{G}_0$ and object articulations $\{\mathbf{G_1}, \cdots, \mathbf{G_B}\}$ using linear blend skinning (LBS) [19, 20]. LBS constrains vertex motion by linearly blending B rigid "bone" transformations with a skinning weight matrix $\mathbf{W} \in \mathbb{R}^{B\times N}$, transforming the canonical

shape into frame t as

$$\mathbf{V}_{i,t} = \mathbf{G}_{\mathbf{0},\mathbf{t}} \left(\sum\nolimits_b \mathbf{W}_{b,i} \mathbf{G}_{b,t} \right) \bar{\mathbf{V}}_i \tag{1}$$

where i is the vertex index, b is the bone index.

We define a set of surface properties for rendering, including vertex 3D coordinates, textures and features, and rasterize them in a differentiable manner [22]. We denote the differentiable rendering procedure of surface properties as $\mathcal{R}(\cdot)$ and omit the shape and motion parameters. To render optical flow, we rasterize and project vertex coordinates in two consecutive frames and compute their 2D displacements [41]. Such renderings are compared against video observations to compute gradients for updating model parameters.

3.2 Video-specific Surface Embedding

Pixel-surface embeddings. We learn pixel and surface embeddings that map corresponding pixels in different frames to the same point on a canonical 3D surface. Intuitively, consider a particular region on the canonical surface mesh that is the "nose" of an articulated human. The surface embedding captures a descriptor for the nose, which can then be matched to pixel-level descriptors at each frame.

Given an input image I_t, the pixel-wise descriptor embedding is represented by a U-Net encoder:

$$\mathbf{F}_{\mathbf{I}}[x, y] = \psi(I_t)[x, y] \in \mathbb{R}^{64}, \tag{2}$$

where $\{x, y\}$ are pixel locations in frame t. The surface embedding is represented by a position-encoded MLP:

$$\mathbf{F}_{\mathbf{S}}(X, Y, Z) = \phi(X, Y, Z) \in \mathbb{R}^{64}, \tag{3}$$

where $\{X, Y, Z\}$ are 3D locations in the canonical space and $\phi(\cdot)$ is a position-encoded MLP with Fourier features [24]. The two embeddings are optimized on test videos such that pixels representing the same surface location in different frames are mapped to the same canonical surface point [19].

Correspondence and soft-argmax regression. Given the pixel and surface embeddings, we construct a per-frame cost volume $D(\mathbf{F}_{\mathbf{I}}, \mathbf{F}_{\mathbf{S}})$ of size $H \times W \times N$ over pixels and surface points (we randomly sample $N = 200$ surface points at each step) by considering their feature distances,

$$D(\mathbf{F}_{\mathbf{I}}, \mathbf{F}_{\mathbf{S}})[x, y, i] = \left\| \mathbf{F}_{\mathbf{I}}[x, y] - \mathbf{F}_{\mathbf{S}}(X_i, Y_i, Z_i) \right\|_2 . \tag{4}$$

Normalizing the cost volume over the surface dimension yields a softmax distribution over possible pixel-to-surface mappings at each pixel location (x, y),

$$\sigma_{(x,y)}[i] = \frac{e^{-D[x,y,i]}}{\sum_j e^{-D[x,y,j]}} . \tag{5}$$

To regress the exact pixel-to-surface correspondences, we take the expectation of 3D point coordinates over the top-K scoring sampled surface points [16, 40],

$$\hat{\mathbf{S}}[x, y] = \mathbb{E}_{\tilde{\sigma}_{(x,y)}[i]} \mathbf{P}_{\mathbf{i}}. \tag{6}$$

where $\mathbf{P}_{\mathbf{i}} = [X_i, Y_i, Z_i]$ is the i-th sampled surface point and $\tilde{\sigma}$ is the re-normalized distribution with the probability of non-top-K surface points set to zero.

In the backward direction, a distribution over possible surface-to-pixel mappings can be constructed for each sampled surface point i by normalizing the cost volume over pixel dimensions. Similarly, the exact surface-to-pixel correspondence can be regressed by taking the expected 2D coordinates over top-K scoring pixel locations.

$$\sigma_i[x, y] = \frac{e^{-D[x,y,i]}}{\sum_{[x,y]} e^{-D[x,y,j]}} . \tag{7}$$

We visualize the bidirectional pixel-surface mapping in Fig. 3.

Relation to keypoints The output of classic keypoint detectors are often represented as a K-channel heatmap over the pixel grid, where K is the number of keypoints. To define dense keypoints, one may increase the number of heatmap channels, which is computationally heavy. In contrast, we represent dense keypoints as lower-dimensional pixel-surface embeddings, which establishes a mapping between pixels and a canonical 3D surface, but also more computational efficient. DensePose [11]

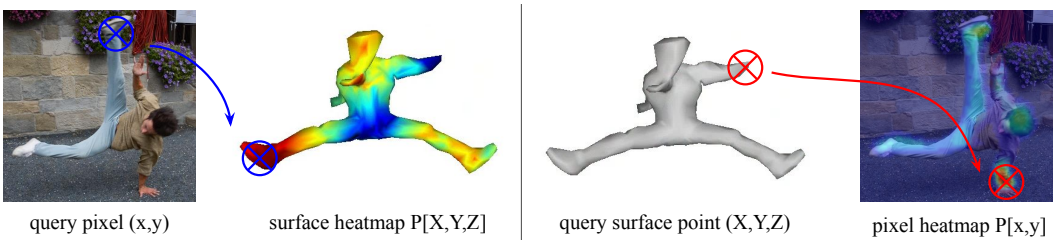

Figure 3: Joint pixel-surface embeddings establish a continuous bijective mapping between between pixels in images and points on a canonical 3D surface. **Left:** Given a query pixel at (x,y), we match it to a set of surface points through the jointly-learned embedding. The matching distribution is used to compute a continuous mapping to the 3D surface. **Right:** Given a query surface point at (X,Y,Z), a matching distribution over the 2D pixel grid can be computed. Warm color indicates high probability of matching.

and CSM [19] use an alternative pixel-to-surface mapping that regresses a (u,v) surface coordinate at every pixel. Our pixel-surface embedding models multimodal uncertainties over keypoints; for example, $\sigma_{(x,y)}[i]$ can capture the fact that a particular pixel matches well to both the left and right ankle, as visualized in Fig. 3, while a regressor tends to "regress" to the mean of the two surface coordinates.

3.3 Learning Embeddings and Articulated Shapes

Next we will introduce the loss functions that enable learning both embeddings and articulated shapes from monocular videos without a pre-defined shape template or annotated correspondence.

Learning pixel-surface embeddings To learn video-consistent and discriminative embeddings, we construct a consistency loss and a contrastive loss, taking advantage of differentiable rendering and geometry cycle consistency.

The *consistency* loss aims to minimize the difference between the top-down rendered surface descriptors and the bottom-up observed pixel descriptors through differentiable rendering:

$$L_{\text{cs}}(\psi, \phi, \bar{\mathbf{V}}, \mathbf{W}, \mathbf{G}) = \sum_{x,y,t} \left|1 - \langle \mathcal{R}(\mathbf{F_S})[x,y,t], \mathbf{F_I}[x,y,t] \rangle\right|, \tag{8}$$

where $\langle , \rangle$ denotes the inner product and $\mathcal{R}(\mathbf{F_S})$ are differentiably rendered surface descriptors. The variables to be optimized include the pixel-surface embeddings (ϕ, ψ), the canonical shape $\bar{\mathbf{V}}$, skinning weights $\mathbf{W}$, and articulations $\mathbf{G}$. This term forces the surface descriptors to be similar to the observed pixel descriptors at the rendered locations, but admits a trivial solution, where all pixel and surface features are the same constant (yielding zero error).

To address this, we add a *contrastive matching* loss [12, 16] that ensures that pixel embeddings match to the surface embedding rendered at the pixel location and do not match to other surface embeddings:

$$L_{\text{ct}}(\psi, \phi) = \sum_{x,y,t} \left\| \mathcal{R}(\bar{\mathbf{V}})[x,y,t] - \hat{\mathbf{S}}[x,y,t] \right\|_2 \tag{9}$$

where $\mathcal{R}(\bar{\mathbf{V}})$ is the rendered 3D surface location and $\hat{\mathbf{S}}[x,y,t]$ is the estimated pixel-to-surface mapping from (6), computed through sampled surface points for computing the softmax distribution $\sigma[i]$. In order for a pixel (x, y) to match to the 3D surface point projecting to that pixel location, the embeddings of other surface points will be pulled away from the pixel embedding.

Learning shape and articulation To better leverage keypoint predictions, we follow prior work of 3D model fitting to construct a cycle-based re-projection error term [3], which pulls the projected 3D keypoints close to the observed ones. Given an input image, we establish a 2D-3D mapping by bottom-up extracting a pixel embedding and matching it to surface embedding. Then, we project the matched 3D surface points to 2D $\pi : \hat{\mathbf{S}} \rightarrow \mathbf{p}$. Different than prior work that defines losses on sparse keypoints, we leverage the pixel embeddings to compute re-projection errors densely on pixels,

$$L_{\text{reproj}}(\psi, \phi, \mathbf{T}, \mathbf{W}, \mathbf{K}, \mathbf{G}) = \sum_{\text{p}ixels} ||\mathcal{R}(\hat{\mathbf{S}}(\mathbf{p})) - \mathbf{p}||_2 \tag{10}$$

where $\mathbf{p}$ is the pixel coordinate. Different from geometric cycle consistency loss [19] that suffers from depth ambiguity along the back-projected ray, our formulation leverages graphics rasterization to automatically handle self-occlusion during joint embedding learning.

Besides reprojection loss, we adopt other inverse rendering losses used in LASR,

$$L_{\text{IG}} = \beta_1 ||\hat{S}_t^i - S_t||_2^2 + \beta_2 \sigma_t ||\hat{\mathbf{u}}_t^i - \mathbf{u}_t||_2 + \beta_3 ||\hat{I}_t^i - I_t||_1 + \beta_4 \textit{pdist}(\hat{I}_t, I_t) \tag{11}$$

where $\{\beta_1, \cdots, \beta_4\}$ are weights empirically chosen, σ_t is the normalized confidence map for flow measurement, and *pdist*$(\cdot, \cdot)$ is the perceptual distance [42] measured by an ImageNet-pretrained AlexNet. The inverse graphics losses ensure the match between rendered and observed optical flow, texture and silhouette images.

Regularization To avoid degenerate shapes, we use mesh Laplacian regularization [15, 41] to enforce the recovered shape to be smooth, and as-rigid-as-possible regularization to enforce the deformation to be locally rigid. Different from prior work [37] that only preserves the length of edges after articulation, we encourage both the area and length of faces to be the same after articulation. The area preserving term is defined as

$$L_{\text{ARAP-area}} = \sum_{i=1}^{|E|} \sum_{j \in N_i} \left| \left| \mathbf{E_i^t} \times \mathbf{E_j^t} \right| - \left| \mathbf{E_i^{t+1}} \times \mathbf{E_j^{t+1}} \right| \right|, \tag{12}$$

where $|E|$ is the number of edges and N_i are the indices of neighbouring edges.

3.4 Neural Implicit Surface Properties

Besides features, we model other surface properties via neural implicit functions defined on a canonical surface. Compared to explicitly defined textures, such continuous implicit representation has the capacity to encode arbitrary amount of details and is empirically easier to optimize.

Neural implicit texture. Textures are defined on mesh surfaces with position-and-image-encoded MLPs to handle multiple instances and appearance variation due to lighting and shadows,

$$\mathbf{C_t} = \mathbf{MLP_{tex}}(\mathbb{F}(\bar{\mathbf{V}}), \psi_{\mathbf{tex}}(\mathbf{I_t})) \in \mathbb{R}^3, \tag{13}$$

where $\psi_{tex}(I_t)$ is a lightweight ResNet that extracts a 64-dimential latent code from an image and $\mathbb{F}$ represents a Fourier encoding used in NeRF [24].

Neural implicit deformation fields. To deal with videos of multiple instances, we model shape variation across instances by a continuous surface deformation field defined on the canonical surface. We represent surface deformation field by a position-and-image-encoded MLP too,

$$\mathbf{V_k} = \bar{\mathbf{V}} + \mathbf{MLP_{shape}}(\mathbb{F}(\bar{\mathbf{V}}), \psi_{\mathbf{shape}}) \in \mathbb{R}^3, \tag{14}$$

where $\bar{\mathbf{V}}_k$ is the rest shape of instance k and ψ_{shape} is a video-specific 64-dimentional latent code.

4 Experiments

We evaluate ViSER in three different scenarios where objects are highly articulating making it challenging to estimate long-range pixel correspondences. First, we consider long human videos with loose clothing and unusual poses. Then, we also evaluate on videos of articulated animals for which accurate shape templates are missing. Finally, we analyze with multiple videos from the same category. All scenarios require jointly establishing long-range correspondences and performing articulated 3D reconstruction at the same time.

4.1 Athletic video reconstruction

Dataset. To evaluate ViSER on long-videos, we construct an athletic video dataset that is challenging due to loose clothing and unusual body poses. It consists of four videos from DAVIS [27] and three ballet videos. All videos are segmented and manually annotated with keypoints following the MSCOCO format [21]. We only use keypoint annotations for evaluation purposes.

Metrics. Due to the lack of ground-truth 3D data for challenging athletic human videos, we use 2D keypoint transfer as a proxy metric [2, 19, 41]. Given any two frames from a video, the goal is to transfer an annotated 2D keypoint from one frame to another. The accuracy is measured by percentage of correctly transferred keypoints over all T(T-1) pairs of frames in a T-frame video. A transferred keypoint is marked as correct when its distance to the ground-truth annotation is lower

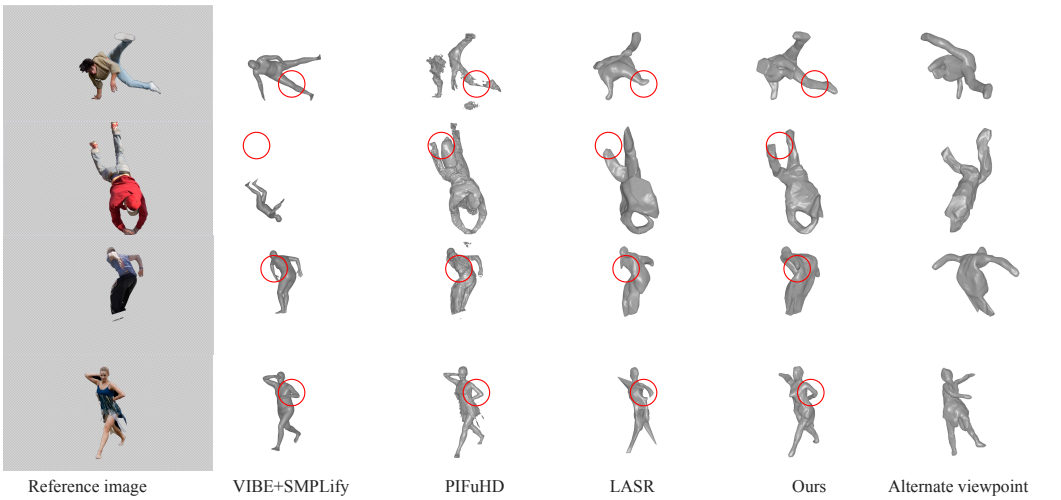

Figure 4: Qualitative comparisons on athletic video shape reconstruction. Compared to methods that heavily rely on shape and pose priors (VIBE+SMPLify and PiFUHD), our method achives comparable performance for common appearance and poses, and does much better on unusual poses such as breakdancers. Please find more results in the supplement.

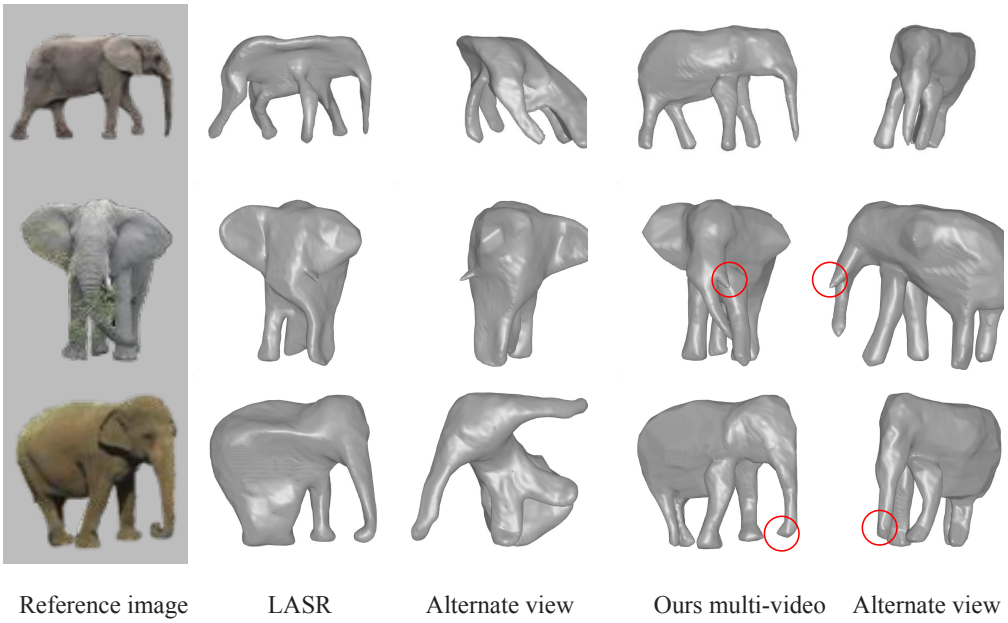

Figure 5: Qualitative comparisons on elephant shape reconstruction. Notice ViSER is able to take advantage of multiple videos to improve the category-level shape reconstruction but also reconstruct instance-specific details (as shown in red circles).

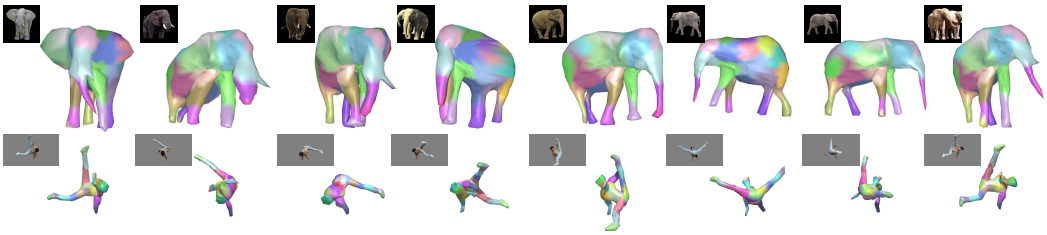

Figure 6: Part segmentation results. Colors are determined by hard-assigning vertices to the closest rigid bones.

Table 1: 2D Keypoint transfer accuracy on athletic videos. Methods with * use keypoint annotations to train. Best results are in bold.

Method	break-1	break-2	dance	parkour	ballet-1	ballet-2	ballet-3	Ave.
*DensePose CSE [25]	56.0	13.2	77.2	**85.9**	45.6	49.0	64.5	55.9
*VIBE+SMPLify [17]	37.1	8.2	70.4	83.8	**55.4**	53.0	**78.8**	55.2
LASR [41]	29.1	18.1	56.6	49.8	44.5	47.4	48.6	42.0
ViSER (Ours)	**70.5**	**22.5**	**80.7**	62.9	52.7	**56.1**	59.9	**57.9**

Table 2: 2D Keypoint transfer accuracy on multiple elephant videos. Best results are in bold.

Method	inner	across
CSE [25]	55.7	51.7
Flow-VCN [40]	51.1	41.2
LASR [41]	57.8	-
ViSER (Ours)	**80.4**	**68.9**

Table 3: 2D Keypoint transfer accuracy on BADJA. Best results are in bold.

Method	camel	dog	cows	horse	bear
CSE [25]	48.8	38.6	63.8	60.2	76.6
Flow-VCN [40]	47.9	25.7	60.7	14.4	63.8
N-NRSfM [31]	67.8	17.9	70.0	8.7	60.2
LASR [41]	**81.9**	65.8	**83.7**	49.3	85.1
ViSER (Ours)	80.1	**73.8**	82.9	**76.3**	**87.3**

than $d_{th} = 0.2\sqrt{|S|}$, where $|S|$ is the area of the ground-truth silhouette [2]. In general, a more accurate reconstruction leads to higher transfer accuracy.

Baselines. To compare with template-based approaches for video human reconstruction, we use VIBE with SMPLify temporal smoothing [17]. To compare with template-free methods, we use LASR [41], which also reconstructs articulated shapes using the same input setting as ours. For reconstruction-based methods, i.e., VIBE+SMPLify and LASR, keypoints are transferred by re-projection, where we back-project the annotated keypoint in the reference frame to the canonical surface, and then project the intersected 3D point to the target frame. We also compare against Densepose CSE [25], which produces dense pixel-to-surface correspondences for a given category, but does not produce 3D reconstructions. To transfer keypoints for Densepose CSE, we compute pixelwise surface mappings for both frames and find the best matching w.r.t.geodesic distance on the surface. We further qualitatively compare againt the SOTA human reconstruction method, PiFUHD, which only produces reconstruction, but not correspondence.

Results. Figure 4 shows visual reconstruction results on sample videos and for different techniques. ViSER estimates reconstructions that are more faithful to the input than the baselines, especially when the humans have unusual poses like in the first two rows. The accurate long-range correspondence enables ViSER to reconstruct finer details than LASR that does not explicitly try to estimate long-range correspondences. We summarize quantitative comparisons in Tab. 1. There is a moderate performance gap between ViSER and template-based methods when the input fits the latter, such as parkour with tight clothing and usual pose. Note that the supervised Densepose CSE and OpenPose methods fail on breakdance videos due to the novel pose, and also do not work well on ballet dancers due to loose clothing. As a result, template-based approaches that rely on accurate pose recognition, such as VIBE [17] fails. In contrast. our method does not suffer from such poor out-of-distribution generalization. Compared to LASR, ViSER can establish longer-range correspondences and achieves higher keypoint transfer accuracy and better 3D reconstruction.

4.2 Reconstructing animals from a video

We use BADJA [2] to evaluate ViSER on animal videos including camel, cow, dog, bear and horse. Similar to the athletic human video dataset, we compare against template-free methods such as LASR and neural-dense-NRSfM (N-NRSfM). Following LASR, **N-NRSfM** [31] trains a video-specific model for object shape, deformation and camera parameters from multi-frame optical flow estimations [8]. We further report performance comparison with dense correspondence methods such as CSE and an optical flow method, VCN. We use the CSE model trained on corresponding animal categories (except that we use the horse model for camel), and the "robust" model of VCN [1], which is the input to our method. As shown in Tab. 3, ViSER achieves better or similar accuracy on all five animal videos compared to LASR and N-NRSfM. While the input optical flow is not robust at estimating long-range correspondences, our method integrates local optical flow to a dense long-range correspondences via a canonical shape, and achieves much better keypoint transfer accuracy. Note

Table 4: Ablation study on keypoint transfer. Best results are in bold.

Method	break-1	elephants-inner	elephants-cross
Full	**70.5**	**80.4**	**68.9**
w/o consistency loss L_{cs} (8)	66.4	80.1	67.2
w/o contrastive loss L_{ct} (9)	36.2	51.3	42.6
w/o reprojection loss L_{reproj} (10)	38.3	80.1	62.5
CSM regression [19]	47.1	77.4	63.3

that CSE performs well for categories it has been trained on, such as cow, horse and bear, but performs poorly on novel animal categories, such as camel and a novel breed of dog.

4.3 Multi-video shape and correspondence

We curate a set of seven videos of different elephants from YTVOS [39] for multi-video shape and correspondence recovery. We treat multiple videos as a single long video with strong appearance changes and shape variations. In the multi-video setup, We evaluate keypoint transfer accuracy on both the same instance (with video frames) and over different instances (across video frames), as denoted by "inner" and "across". Quantitative results in Tab. 2 shows that ViSER is more accurate than the baseline methods in both cross-video keypoint transfer and inner-video keypoint transfer by a large margin, without using any keypoint annotations or pre-defined shape templates. Fig. 5 show visual result comparisons. While LASR recovers the visible surfaces in a video, it cannot infer the invisible parts. In contrast, our method is able to take advantage of multiple videos from the same category and produce a much better shape reconstruction. Note that LASR can not handle multiple videos as it requires optical flow computed between every adjacent frame pairs. ViSER, on the other hand, also uses correspondences via estimated 3D shape there by allowing the use of multiple videos even when the optical flow is missing across videos.

4.4 Part discovery and alblations

Part discovery. ViSER can discover detailed 3D part segmentation without any manual annotation, as shown in Fig. 6. After training either on a collection of videos or a long video, ViSER can segment the 3D shape into meaningful parts, such as the trunk of the elephants and the feet of the dancer.

Ablation study. We perform an ablation study on rbreak-1 and elephants, as shown in Tab. 4. Removing the consistency loss slightly degrades the performance. Without the contrastive loss, the pixel-surface embedding converges to a trivial solution with a significant decrease of accuracy. Removing the re-projection loss leads to much lower keypoint transfer (KPT) accuracy on the breakdance-1 sequence and cross-video KPT accuracy on the elephant videos. Likely the surface reprojection loss plays an important role in learning correct articulation that follows the bottom-up dense keypoint predictions. This may effectively avoid the local minimum issue for the differentiable rendering optimization. Finally, replacing the pixel-surface embedding with direct CSM regression [19] does not reason about distribution of possible matches and results in worse performance.

Broader impact. ViSER has many potential applications, *e.g.*, in robotics, AR/VR, and film industry, but may be used for malicious purposes, *e.g.*, producing fake videos or extracting bio-metric information without prior consent. ViSER is only suitable to offline applications as it takes about several hours to process a 80-frame video on one NVIDIA P100 GPU.

5 Conclusions

We have introduced ViSER, a method to reconstruct articulate shapes, dense trajectories, and object parts from monocular videos. ViSER establishes long-range correspondence by matching 2D pixels to a canonical 3D mesh via learned video-specific surface embeddings. Experimental results show that ViSER, without a template shape or keypoint annotations, compares favorably against prior work on challenging human and animal videos. ViSER shows that it could be fruitful to reconstruct articulate shapes for categories in the wild, and we hope to see more work in this direction.

References

[1] Robust vision challenge. www.robustvision.net, 2020.

[2] B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla. Creatures great and smal: Recovering the shape and motion of animals from video. In *ACCV*, pages 3–19. Springer, 2018.

[3] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016.

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, volume 2, pages 690–696. IEEE, 2000.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *ECCV*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.

[6] T. Darrell and A. P. Pentland. Cooperative robust estimation using layers of support. *TPAMI*, 17(5): 474–487, 1995.

[7] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, pages 834–849. Springer, 2014.

[8] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *IJCV*, 2013.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. CVPR*, pages 3354–3361. IEEE, 2012.

[10] P. F. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR*, pages 3065–3072. IEEE, 2011.

[11] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, pages 7297–7306, 2018.

[12] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006.

[13] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761. IEEE, 1993.

[14] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 41(1):190–204, 2017.

[15] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[16] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, pages 66–75, 2017.

[17] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, June 2020.

[18] C. Kong and S. Lucey. Deep non-rigid structure from motion. In *ICCV*, pages 1558–1567, 2019.

[19] N. Kulkarni, A. Gupta, D. F. Fouhey, and S. Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, pages 452–461, 2020.

[20] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, 2000.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[22] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *ICCV*, Oct 2019.

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

[24] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, 2020.

[25] N. Neverova, D. Novotny, V. Khalidov, M. Szafraniec, P. Labatut, and A. Vedaldi. Continuous surface embeddings. In *NeurIPS*, 2020.

[26] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

[27] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016.

[28] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: A new representation for video editing. In *SIGGRAPH*, pages 1–11, 2008.

[29] Z. Ren, O. Gallo, D. Sun, M.-H. Yang, E. B. Sudderth, and J. Kautz. A fusion approach for multi-frame optical flow estimation. In *WACV*, pages 2077–2086. IEEE, 2019.

[30] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80(1): 72, 2008.

[31] V. Sidhu, E. Tretschk, V. Golyanik, A. Agudo, and C. Theobalt. Neural dense non-rigid structure from motion with latent space constraints. In *ECCV*, pages 204–222. Springer, 2020.

[32] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2): 189–210, 2008.

[33] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, June 2018.

[34] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, pages 438–451. Springer, 2010.

[35] Z. Teed and J. Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[36] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.

[37] S. Tulsiani, N. Kulkarni, and A. Gupta. Implicit mesh reconstruction from unannotated image collections. In *arXiv*, 2020.

[38] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, Sept. 1994.

[39] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *ECCV*, pages 585–601, 2018.

[40] G. Yang and D. Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, pages 794–805, 2019.

[41] G. Yang, D. Sun, V. Jampani, D. Vlasic, F. Cole, H. Chang, D. Ramanan, W. T. Freeman, and C. Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021.

[42] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please see Section 4.4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please see Section 4.4.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release the code upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 4 and the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Section 4.4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Please see Section 4.
   (b) Did you mention the license of the assets? [Yes] Please see Section 4.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]