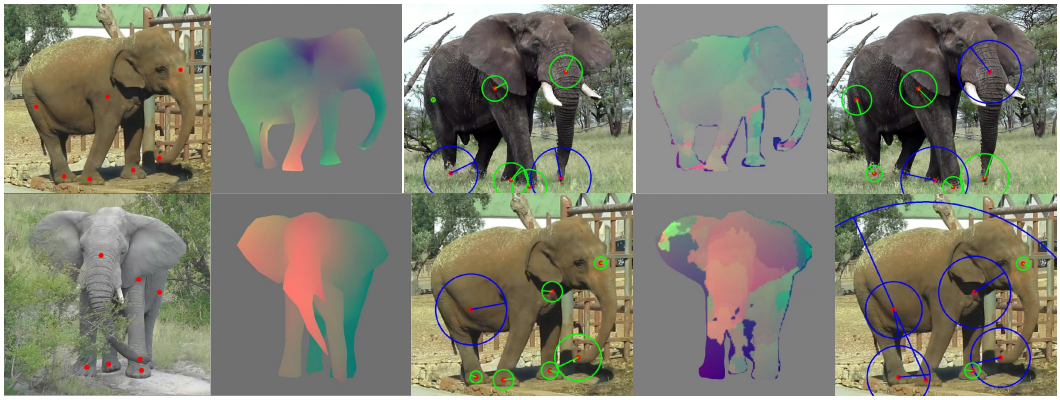
Supplementary Materials for ViSER: Video-Specific Surface Embeddings for Articulated 3D Shape Reconstruction

A Appendix

A.1 Keypoint Transfer Evaluation

We visualize keypoint transfer evaluation procedure in Fig. 1. Keypoints are annotated for meaningful body parts of human and elephants in the MSCOCO format. For human, we annotate fifteen points including "nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, mid hip, right hip, right knee, right ankle, left hip, left knee, and left ankle". For elephants, we annotate eleven keypoints including two keypoints for each leg, two keypoint for nose, and one keypoint for tail.



reference image | correspondence field (ViSER) | transferred keypoints (ViSER) | correspondence field (CSE) | transferred keypoints (CSE)

Figure 1: Visualization of keypoint transfer results. Red dots indicate annotated keypoints, green circle indicates successful transfer, and blue circle indicates unsuccessful transfer. We transfer keypoints in the reference image with inferred dense correspondence fields as discussed in Sec. 4.1. Such correspondence fields can be established between any two frame in a video, or a set of videos by re-projection via a canonical 3D model. The inferred correspondence field of ViSER is crisper and generally more accurate than CSE [5].

A.2 Implementation Details

Surface property rendering As briefly mentioned in Sec. 3.1, we render surface properties using a differentiable renderer, SoftRas [3]. Specifically, we have

$$\hat{I}_t = \mathcal{R}(\mathbf{C_t}; \mathbf{V_t}, \mathbf{F}) \tag{1}$$

where $\hat{I}_t$ is the rendered raw pixels, $\mathcal{R}(\cdot)$ is the rasterization function, $\mathbf{C_t}$ is the texture sampled at each mesh vertex position using Eq. (13) of the main text, $\mathbf{V_t}$ is the articulated vertices defined in Eq. (1) of the main text, and $\mathbf{F}$ is the topology of a mesh. We refer readers to LASR [12] for rendering equations of optical flow.

Surface coordinate-based MLPs As discussed in Sec. 3.2 and Sec. 3.4, we use coordinate-based MLPs with Fourier encoding to represent canonical surface properties, including surface features $\mathbf{F}$, texture $\mathbf{C_t}$, and instance-specific shape deformation $\mathbf{\Delta V_k}$.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Specifically, we follow NeRF [4] to encode the input (X, Y, Z) coordinates (but not viewing direction) with a set of sine and cosine functions of increasing frequency before passing into an MLP regressor, which is shown useful for learning high-frequency functions from low-dimensional inputs [10].

To encode time-varying texture (due to lighting and shadows), we follow Pixel-NeRF [14] to further learn an image-dependent latent code that modulates the intermediate features of the video-specific texture MLPs. Similarly, to learn instance-specific shape deformation, we learn an instance-specific latent code to modulate the category-specific deformation MLPs.

Additional regularization on bones Following LASR [13], we use a set of Gaussian "bones" in the canonical space to represent skinning weights. However, we find the bones tend to move outside the surface. We posit that the skinning weights are ambiguous to optimize from limited video observations. In practice, we force the bones to stay around the object surface by minimizing the distance between bone centers and sampled surface points. The distance is measured with Sinkhorn divergence [1], which interpolates between optimal Wasserstein and kernel distances.

Incremental optimization Similar to classic SfM, we reconstruct a long video sequence (>50 frames) in an incremental manner [8]. Specifically, we use an initial set of frames to start the optimization. The initial set is selected based on the following heuristics: (1) The flow magnitude between adjacent frames are large enough to ensure motion parallax; (2) The number of frames is between 10 and 30. Then, we gradually add in new frames, until all the video frames are added and optimize them jointly. Empirically, simultaneously optimizing all the video frames yields worse results, possibly due to the noise in stochastic gradient updates, which may be reduced with a larger batch size and more GPUs.

A.3 Comparison to Neural Scene Flow Fields [2]

Related to our problem setup, some recent methods reconstruct dynamic neural radiance fields (NeRF) from a monocular video [2, 6, 7, 11] by differentiable volume rendering, and achieve promising results for novel view synthesis and depth estimation in dynamic scenes.

However, such methods may not work well when the objects exhibit large motion, such as root body rotations. To illustrate this point, we compare with Neural Scene Flow Fields [2] (NSFF), which also use two-frame optical flow as inputs. We ran the public code of NSFF on the DAVIS "dance-twirl" sequence, and the results are shown in Fig. 2.

Specifically, we use COLMAP [8, 9] to estimate camera parameters with regard to the static background (with foreground objects removed). For the "dance-twirl" sequence, COLMAP registers all 90 frames and reconstruct a reasonable background. Then we train NSFF using 4 GPUs for 280k iterations. To extract the reconstructed surface, we sample points from a 256x256x256 grid in the canonical space and run marching cubes. As a result, we find that although NSFF can "overfit" to the input image and optical flow, the extracted surface of the dynamic human is completely off – with streaks connecting the background to the foreground. We hypothesize that one of the possible reasons for NSFF to fail is the lack of long-range correspondence. In contrast, our method is able to deal with root body rotation due to the estimation of long-range correspondences via canonical surface features, as well as the usage of blend-skinning model which regularizes the motion.

Table 1: Table of notations.

Symbol	Description
	Constants
T	Number of frames in the input video
M	Number of faces in the mesh
N	Number of vertices in the mesh
B	Number of bones for linear blend skinning (LBS)
$\mathbf{F}$	Topology of the mesh
$\boldsymbol{\beta}$	Weights of losses
	Input Measurements
I_t	Input RGB image at time t
S_t	Input or measured object silhouette image at time t
u_t^+	Input or measured forward optical flow map from time t to $t+1$
u_t^-	Input or measured backward optical flow map from time t to $t-1$
	Renderings
$\hat{I}_t$	Rendered color image of the object at time t
$\hat{S}_t$	Rendered object silhouette image at time t
$\hat{u}_t^+$	Rendered forward optical flow map of the object from time t to $t+1$
$\hat{u}_t^-$	Rendered backward optical flow map of the object from time t to $t-1$
	Shared Model Parameters
$\bar{\mathbf{V}}_i$	Position of the i-th mesh vertex of the rest shape
$\mathbf{W}$	Skinning weights matrix
$\psi_p(I_t)$	Weights of the ResNet-18 pose network
$\psi_{tex}(I_t)$	Weights of the ResNet-18 that produces texture code ω_t
$\psi_e(I_t)$	Pixel embedding, parameterized by a 2D U-Net
$\phi_e(X,Y,Z)$	Surface embedding, parameterized by a coordinate-MLP
$\phi_{tex}(X,Y,Z,\omega_t)$	Surface texture, parameterized by a coordinate-MLP
τ	Temperature parameter for softmax matching distribution over surface points, Eq. (5)
	Time-Varying Model Parameters
ω_t	A texture code associated with each image t
$\mathbf{V_t}$	Position of mesh vertices at time t
$\mathbf{C}_{i,t}$	Color of mesh vertices at time t
$\mathbf{K_t}$	Intrinsic matrix of a simple pinhole camera (with zero skew and square pixel) at time t
$\mathbf{G_{0,t}}$	Object root body SE(3) transformation at time t
$\mathbf{G_{1\ldots B,t}}$	Bone SE(3) transformations at time t
	Additional Parameters for Multi-Video Optimization (Sec. 3.4)
α_k	A shape code associated with each video k
$\phi_{shape}(X,Y,Z,\alpha_k)$	Video-specific shape deformation from a canonical shape, parameterized by a coordinate-MLP

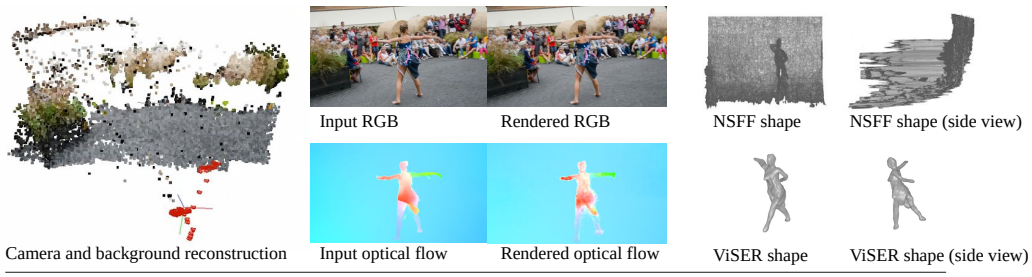


Figure 2: Comparison to Neural Scene Flow Fields (NSFF) on dance-twirl sequence. Data preparation: We run COLMAP to reconstruct the background and register cameras (in red) of all frames with regard to the reconstructed background. NSFF-training: We train NSFF for 280k iterations on 4 GPUs, and observe that NSFF is able to "overfit" to the input image and optical flow. Surface extraction: we extract surface with marching cubes at a 256x256x256 sampled grid. For NSFF, we observe that the extracted surface of the dynamic human is completely off – with streaks connecting the background to the foreground. In contrast, ViSER is able to correctly reconstruct the dancer.

References

[1] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.

[2] Z. Li, S. Niklaus, N. Snavely, and O. Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

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

[4] 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.

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

[6] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[7] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2020.

[8] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[9] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[10] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020.

[11] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021.

[12] 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.

[13] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392. IEEE, 2011.

[14] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.