Learning Motion-Dependent Appearance for High-Fidelity Rendering of Dynamic Humans from a Single Camera

Jae Shin Yoon Duygu Ceylan Tuanfeng Y. Wang
Jingwan Lu Jimei Yang Zhixin Shu Hyun Soo Park

University of Minnesota
Adobe Research

Introduction

- Models a sequence of body meshes (e.g., SMPL) to renders.
- Trained per-subject on **limited** amount of dataset (15-60s video) while mitigating overfitting by utilizing *equivariance*.
- While most other models only consider pose to model 3D (including secondary motion, e.g., clothing dynamics) body, we factor in spatial and temporal derivatives—i.e., velocity).
- 3D features are rearranged to a UV map, which is invariant to displacement of vertices, does not suffer from ambiguity like that of 2D projection, and suited for 2D convolution due to locality.

Introduction

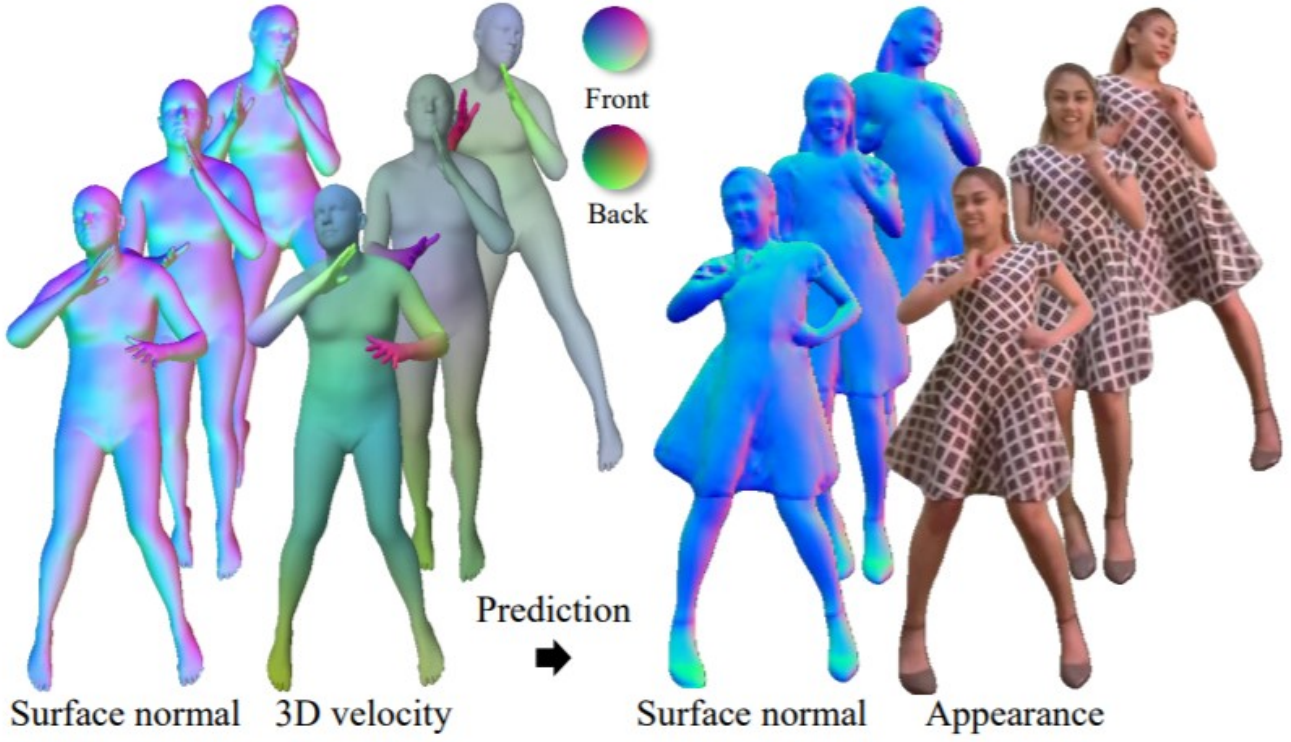


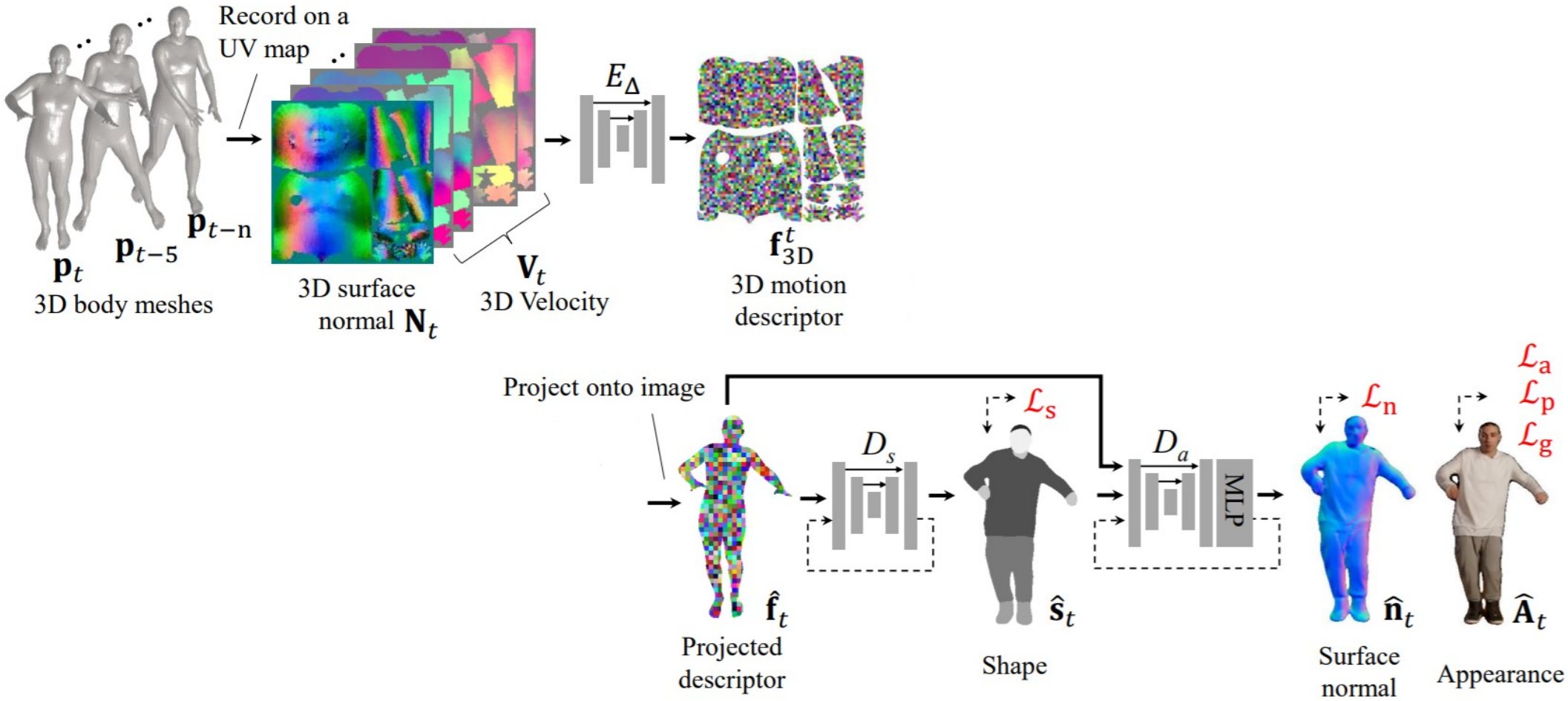
Figure 1: Subject-specific body model sequence parsed to generate surface normals and 3D velocities to be used as input

Related Work

- Model-based approaches: Leverage 3D shape models
- Retrieval-based approaches: Synthesize image; neural rendering, generative networks:
  - Most existing approaches do not model time-varying secondary motion
  - *Real-time deep dynamic characters* [19]: Utilizes a motion cue, requires pre-learned person-specific 3D template model
  - *Monocular human animation with neural dynamic appearance synthesis, 2021* [62]: Seq. of dense surface parametrization to motion features that synthesize dynamic appearances using StyleGAN

Method: Overview

- Given a monocular video of a person in motion and the corresponding 3D body fit estimates, learn a motion representation to describe time-varying appearances of the secondary motion—inputting different person shape possible during testing?



Method

- Human rendering → learning a representation via a feature encoder-decoder framework:

$$\boldsymbol{f} = E(\boldsymbol{p}), \boldsymbol{A} = D(\boldsymbol{f})$$

p: posed body representation

f: per-pixel features

A: appearance ([0, 1], w×h×3)

- Problem: both encoder and decoder must memorize every appearance of corresponding pose, while data is limited.

Method: Encoder: Equivariant 3D Motion Descriptor (cont.)

- To address this challenge, we can use an equivariant geometric transformation s.t.:

$$E(W\,\boldsymbol{x}) = W\,E(\boldsymbol{x})$$

i.e., $\boldsymbol{f} = W\,\boldsymbol{f_0} = W\,E(\boldsymbol{p_0}) = W\,E(W^{-1}\,\boldsymbol{p}), A = D(\boldsymbol{f}) = D(W\,\boldsymbol{f_0})$

where W can be modelled as a function that transforms a zero-pose to a desired pose, i.e., an LBS matrix (deterministic).

- We can see that since $\mathbf{f_0}$ is constant the encoder does not need to be learned in this naïve approach.

Method: Encoder: Equivariant 3D Motion Descriptor (cont.)

- However we should factor in the time-varying appearance for the secondary motion as follows:

$$\mathbf{f} = E\left(\mathbf{p}, \frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{p}}{\partial t}\right) \approx E(\mathbf{p}) + E_\Delta\left(\frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{p}}{\partial t}\right)$$

$$\iff \mathbf{f}_0 = \underbrace{E(\mathcal{W}^{-1}\mathbf{p})}_{\text{const.}} + E_\Delta\left(\mathcal{W}^{-1}\frac{\partial \mathbf{p}}{\partial x}, \mathcal{W}^{-1}\frac{\partial \mathbf{p}}{\partial t}\right)$$

- Spatial and temporal derivatives correspond to 3D surface normal and surface velocities respectively. 3D representation of the posed body can be derived: $\mathbf{f}_{3\mathrm{D}} = E_\Delta(\mathcal{W}^{-1}\mathbf{N}, \mathcal{W}^{-1}\mathbf{V}), \;\; \mathbf{A} = D(\Pi\mathcal{W}\mathbf{f}_{3\mathrm{D}})$
- **N** and **V** are surface normals and instantaneous velocities for m vertices—# of verts representing body—which can be mapped to UV coordinates where the positions of the vertices—i.e., UV mapping—are fixed, by interpolating the values of mapped vertices on the UV surface for pixels between the mapped vertices.

Method: Encoder: Equivariant 3D Motion Descriptor

- Because N and V are in UV coordinates, 2D CNN can be applied for our *residual encoder*, helping capture relationship btw. neighbouring body parts.
- $\mathbf{f}_{3D} \in \mathbb{R}^{m \times d}$ is spatially aligned to the UV map.
- Expressing feature map ($\mathbf{f}_{3D}$) in 3D improves *discriminativity*.
- $\mathbf{f}_{3D}$ is first “posed” by multiplying with W then projected to image space via Π—transports features defined in UV space to image plane via coordinate tfm—before being fed to decoder.

Method: Multitask Compositional Decoder

- Decoder is multi-part compositional: $D = D_a \circ D_s$ (better than training end-to-end according to ablation study).
- D_s learns dynamics of 2D shape, whose output is fed to D_a along with the projected features which in turn outputs appearance and surface normals, both in image coordinates.
- Each intermediate representation receives its own supervision signals resulting in multi-task learning. Output of D_s are essentially semantic segmentation, which is supervised for that matter.
- Recurrence: each module modelled as an **autoregressive network** allowing to learn dynamics rather than memorizing pose-specific appearance.

Method: Loss

- $$\mathcal{L} = \sum_{\mathbf{P},\mathbf{A}\in\mathcal{D}} \mathcal{L}_\mathrm{a} + \lambda_\mathrm{s}\mathcal{L}_\mathrm{s} + \lambda_\mathrm{n}\mathcal{L}_\mathrm{n} + \lambda_\mathrm{p}\mathcal{L}_\mathrm{p} + \lambda_\mathrm{g}\mathcal{L}_\mathrm{g}$$
- Loss terms: appearance, shape, surface normal, perceptual similarity, and generative adversarial losses (**P**: GT 3D pose, **A**: GT appearance):

$$\mathcal{L}_\mathrm{a}(\mathbf{P},\mathbf{A}) = \|\widehat{\mathbf{A}} - \mathbf{A}\|,$$
$$\mathcal{L}_\mathrm{s}(\mathbf{P},\mathbf{A}) = \|\widehat{\mathbf{s}} - S(\mathbf{A})\|,$$
$$\mathcal{L}_\mathrm{n}(\mathbf{P},\mathbf{A}) = \|\widehat{\mathbf{n}} - N(\mathbf{A})\|,$$
$$\mathcal{L}_\mathrm{p}(\mathbf{P},\mathbf{A}) = \sum_i \|VGG_i(\widehat{\mathbf{A}}) - VGG_i(\mathbf{A})\|,$$
$$\mathcal{L}_\mathrm{g}(\mathbf{P},\mathbf{A}) = \mathbb{E}_{S(\mathbf{A}),\mathbf{A}}[\log(D^\star(S(\mathbf{A}),\mathbf{A})] + \mathbb{E}_{S(\mathbf{A}),\widehat{\mathbf{A}}}[\log(1 - D^*(S(\mathbf{A}),\widehat{\mathbf{A}})]$$

- *D** is PatchGAN discriminator [21], which validates plausibility of synthesized image condisioned on the shape mask—is it enough that L_g is being minimized? What about the discriminator? OK if *D** pretrained?

Training

- Adam optimizer, lr=10^{-3}.
- Given an input video of ~10K frames (dance videos from YouTube and seq. from prior work) train for 72 hours using 4 NVIDIA V100 GPUs with batch size of 4.
- Body surface normals in current frame, body surface velocities in the past t = 10 frames.
- UV map size: 128×128, Output image size: 512×512
- Pytorch3D differentiable rendering layers

Results: Qualitative

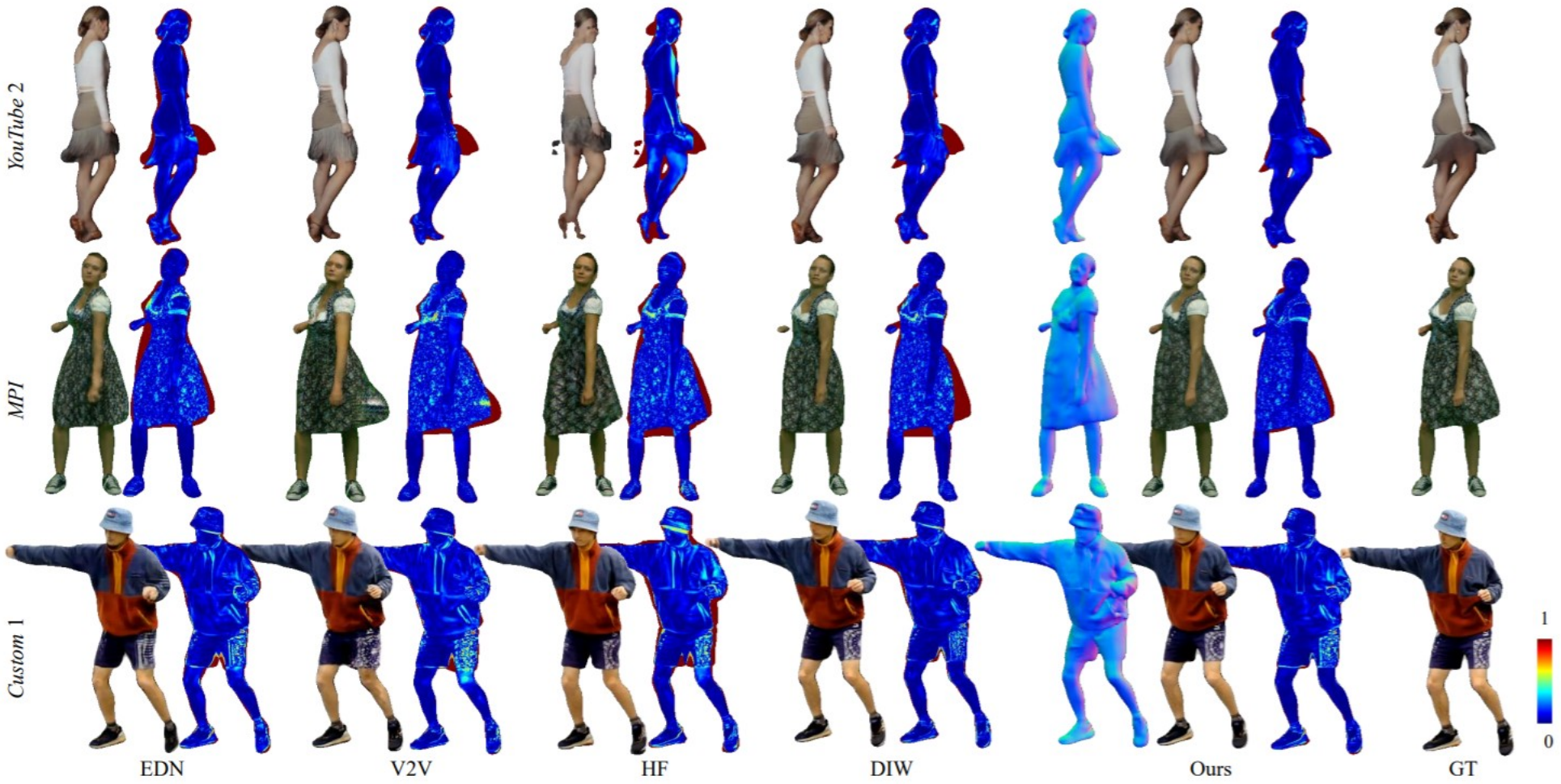


Figure 5. We compare our method to several baselines (EDN [7], V2V [61], HFMT [25], DIW [62]) on various sequences. For each example, we show the ground truth (GT) target appearance, the synthesized appearance by each method, and a color map of the error between the two. For our method, we also visualize the predicted surface normal.

Results: Quantitative

Method	*YouTube* 1 (6K)	*YouTube* 2 (10K)	*YouTube* 3 (4K)	*MPI* (10K)	*Custom* 1 (15K)	*Custom* 2 (15K)	Avg.
EDN [7]	0.954 / 3.06 / 0.356	0.943 / 4.39 / 0.465	0.871 / 6.23 / 0.467	0.824 / 4.59 / 0.287	0.916 / 5.26 / 0.450	0.928 / 5.06 / 0.423	0.906 / 4.76 / 0.408
V2V [61]	0.960 / 2.23 / 0.235	0.958 / 3.33 / 0.405	0.880 / 4.47 / 0.401	0.824 / 3.58 / 0.298	0.935 / 3.52 / 0.306	0.943 / 4.15 / 0.385	0.916 / 3.54 / 0.338
HFMT [25]	0.944 / 4.19 / 0.412	0.923 / 6.63 / 0.775	0.862 / 7.16 / 0.456	0.826 / 5.03 / 0.291	0.905 / 6.24 / 0.321	0.915 / 6.63 / 0.390	0.895 / 5.98 / 0.440
DIW [62]	0.966 / 2.21 / 0.275	0.960 / 3.03 / 0.370	0.894 / 4.69 / 0.396	0.825 / 2.94 / 0.359	0.939 / 3.23 / 0.304	0.944 / 3.95 / 0.412	0.921 / 3.34 / 0.336
Ours	0.973 / 2.01 / 0.240	0.964 / 2.83 / 0.338	0.897 / 4.50 / 0.412	0.825 / 2.82 / 0.203	0.942 / 3.12 / 0.279	0.946 / 3.81 / 0.404	0.925 / 3.18 / 0.312

Table 1. Quantitative results. The number of training frames in each sequence is given in the top row. The thee numbers are the SSIM (↑), LPIPS (↓)×100, and tLPIPS (↓)×100 metrics, respectively. The red represents the best performer, and the blue second best.

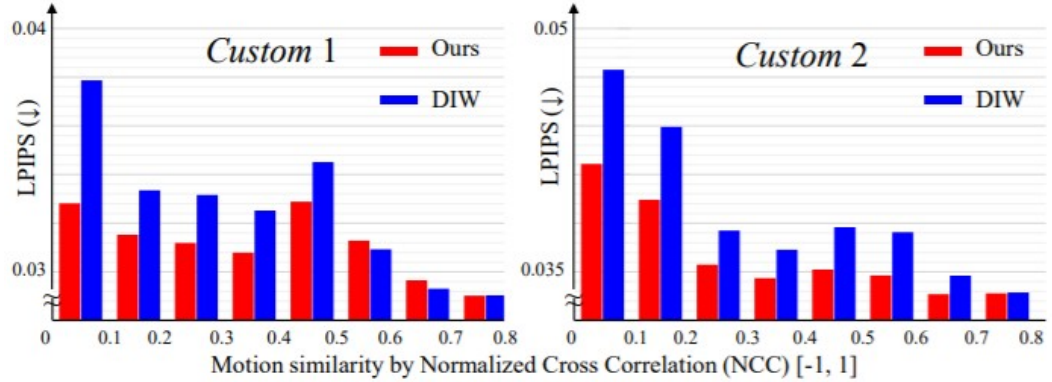


Figure 6. Perceptual quality of a synthesized image over motion similarity between training and testing sequences.

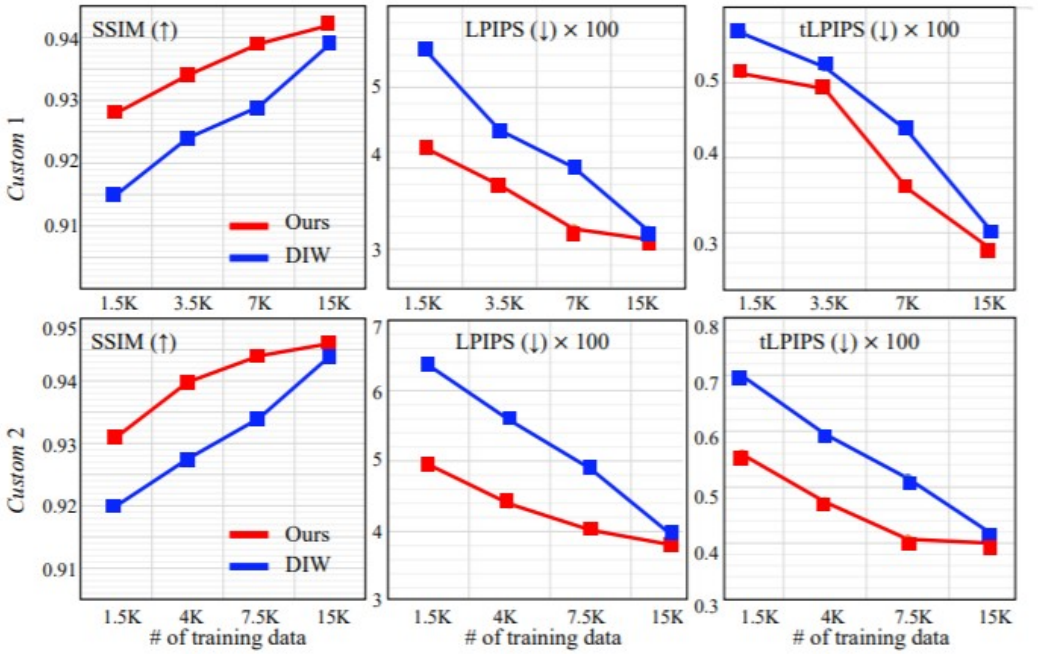


Figure 7. Performance depending on the amount of training data.

DIW: Dance in the wold [62], a different model used as baseline

Applications

Motion transfer & background composition

Bullet time effect (novel view synthesis)

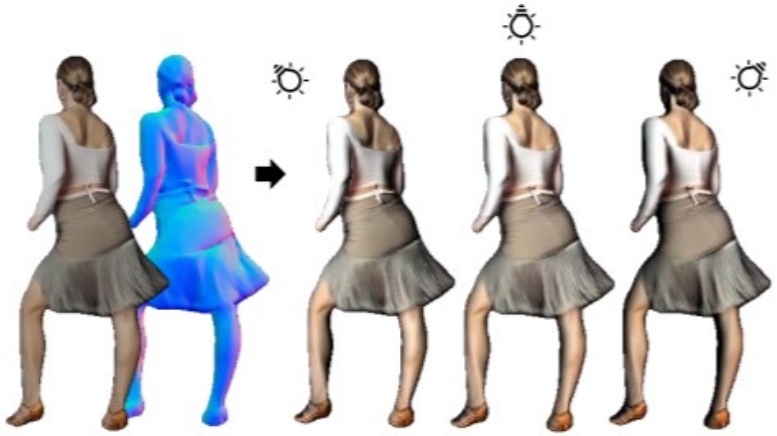


Image relighting

- Motion transfer via transferring joint locations
- Bullet-time effect by rotating 3D body
- Relighting using surface normal prediction