Towards Learning a Realistic Rendering of Human Behavior

Patrick Esser*, Johannes Haux*, Timo Milbich*, Björn Ommer

HCI, IWR, Heidelberg University

Abstract. Realistic rendering of human behavior is of great interest for applications such as video animations, virtual reality and gaming engines. Commonly animations of persons performing actions are rendered by articulating explicit 3D models based on sequences of coarse body shape representations simulating a certain behavior. While the simulation of natural behavior can be efficiently learned, the corresponding 3D models are typically designed in manual, laborious processes or reconstructed from costly (multi-)sensor data. In this work, we present an approach towards a holistic learning framework for rendering human behavior in which all components are learned from easily available data. To enable control over the generated behavior, we utilize motion capture data and generate realistic motions based on user inputs. Alternatively, we can directly copy behavior from videos and learn a rendering of characters using RGB camera data only. Our experiments show that we can further improve data efficiency by training on multiple characters at the same time. Overall our approach shows a new path towards easily available, personalized avatar creation.

1 Introduction

Recently there has been great progress in the field of generating and synthesizing images [11, 18, 42, 14, 34, 23, 30] and videos [35, 33] which is important for applications such as image manipulation, video animation and rendering of virtual environments. Over the past years, in particular the gaming industry eagerly improved their customer experience by developing more and more realistic

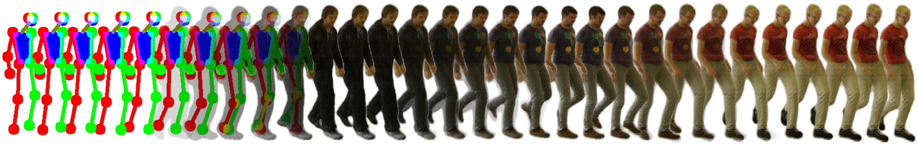

Fig. 1. Our method can render target persons in a wide variety of behaviors. The desired behavior can be either generated based on inputs from a game controller or copied from a video.

* equal contribution

3D gaming engines. These progressively push the level of detail of the rendered scenes, with an emphasis on natural movements and realistic appearances of the in-game characters. Characters are typically rendered with the help of detailed, explicit 3D models, which consist of surface models and textures, and animated using tailored motion models to simulate human behavior and activity.

Recent work [13] has shown that it is possible to learn natural human behaviour (e.g. walking, jumping, etc.) from motion capture data (MoCap) of human actors. On the other hand, designing a realistic 3D model of a person is still a laborious process. Traditionally, explicit 3D models of the shape are manually created by specialized graphic designers, followed by the design of textures reflecting a person's individual appearance which is eventually mapped onto the raw body model. In order to circumvent this laborious design process, passive 3D reconstruction methods can be utilized [3, 12, 1, 5]. While these methods are able to achieve impressive results, they rely on data recorded by costly multi-view camera settings [7, 5, 9, 1], depth sensors [15, 26, 29, 37] or even active 3D laser scans [20].

Given the tremendous success of generative models [11, 18, 41, 17, 14] in the era of deep learning, the question arises, why not also learn to generate realistic renderings of a person, instead of only learning its natural movements? By conditioning the image generation process of a generative model on additional input data, mappings between different data domains are learned [42, 14, 16], which, for instance, allows for controlling and manipulating object shape, turning sketches into images and images into paintings. Thus, by being able to condition a rendering of a person on its body articulation, the laborious 3D modeling process can be circumvented by directly learning a mapping from the distinct postures composing human behaviors onto realistic images of humans.

In this work, we propose an approach towards a completely data driven framework for realistic control and rendering of human behavior which can be trained from Motion Capture data and easily available RGB video data. Instead of learning an explicit 3D model of a human, we train a conditional U-Net [28] architecture to learn a mapping from shape representations to target images conditioned on a latent representation of a variational auto-encoder for appearance. For training our generative rendering framework, we utilize single-camera RGB video sequences of humans, whose appearance is mapped onto a desired pose representation in the form of 2D keypoints. Our approach then enables complete control of a virtual character. Based on user inputs about desired motions, our system generates realistic movements together with a rendering of the character in a virtual environment. Since our rendering approach requires only example videos of the character, it can be easily used for personalized avatar creation. Furthermore, because it operates on 2D keypoints, it can also be used for video reenactment where the behavior shown in a video is copied.

We evaluate our model both quantitatively and qualitatively in an interactive virtual environment and for simulating human activity from recorded video data.

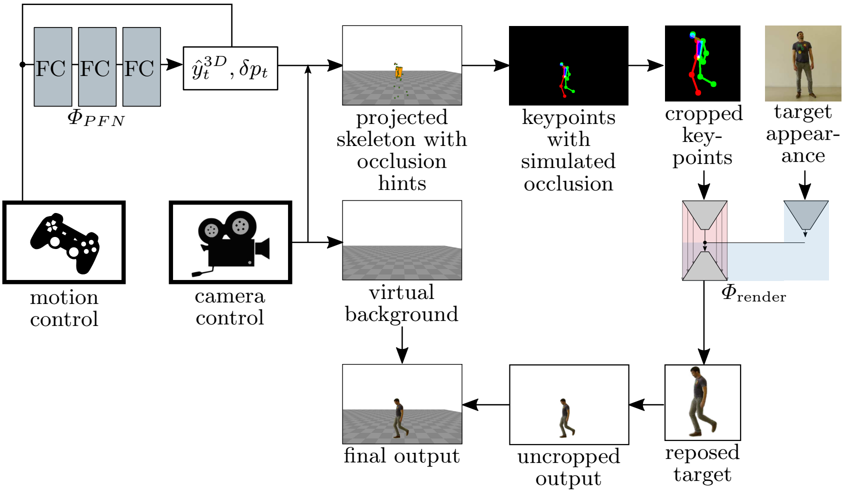

Fig. 2. Our pipeline for controllable synthesis of behavior: Motion controls obtained from a game controller are used by Φ_{PFN} to update a 3D skeleton. Combined with a user controllable camera, we obtain a 3D projection of the skeleton with estimates on keypoint visibility as well as a rendered image of the virtual environment. Using the projected skeleton, we compute a stickman representation compatible with Φ_{render}. The remaining pipeline is similiar to Fig. 3 except that we do not need to perform inpaininting.

2 Related Work

There is a large corpus of works available for tackling the problem of creating realistic 2D renderings of 3D objects and persons, as well as their natural animation. In most cases this is still a laborious and manual process.

3D Modeling: The common approach for learning 3D models from visual data of real objects or humans is a three-dimensional reconstruction of the object of interest based on a collection of input images obtained from different camera settings. Many works utilize multi-view camera recordings of their subjects in combination with specialized shape models describing the human body configuration [22, 3, 43, 12] or without additional shape priors [7, 5, 9, 1]. While these approaches result in impressive reconstructions, obtaining the required input data is costly and asks for specialized equipment and recording settings.
3D scanners and cameras with RGB-D sensors like the KinectFusion system [15, 26] use depth information to increase the level of detail of the resulting 3D models [29, 20, 37]. However, again specialized equipment is needed and extra information used. In contrast, our approach is able to learn renderings of a subject from easily obtained RGB camera recordings.
There are only a few works which operate on monocular image data to learn 3D models. Moreover, these works typically neglect the individual appearance of

the test subjects by modeling only posture skeletons [40, 31] or a fixed, neutral template appearance [24, 27]. Only the approach of Alldieck et al. [2] additionally learns the individual subject appearances and body shapes. However, their method depends on the SMPL model [22] which was trained on a large set of 3D body scans, while our rendering model is solely trained on easily available 2D input data.

Conditional Image Generation: Generative models [11, 18] offer an orthogonal avenue to the task of rendering in-game characters. Instead of an explicit 3D model of a test subject, such models are able to generate images from a latent space and allow for interpolating between the training images such as viewpoint and subjects [32, 17]. Moreover, conditioning the image generation on additional input data (such as class label, contours, etc.) allows for mappings between the conditioning and target domain [25, 14, 41] and thus grants control over the generative process. Similarly to [10, 23, 30, 4, 19] we condition our generative model on pose information.

3 Method

The degree of realism which can be obtained when generating renderings of a certain behavior by a given test subject hinges on two crucial components: *(i)* a realistic appearance model considering both the shape and texture of the person and *(ii)* a motion model which is able to capture the dynamics of the behavior and is able to control the deformation of the appearance model. Furthermore, such a motion model must be able to describe the distinct body articulations involved while performing an action and needs to simulate the natural transitions between them, i.e. the actual behavior.

In our framework, visualized in Fig. 2, both components are represented by deep neural networks, which are trainable from Motion Capture data and easily available RGB video data, respectively. Since the description of human shape and motion is a well-studied problem with efficient methods available, we integrate a pose estimation algorithm, OpenPose [6], for encoding human articulation and a motion model, PFNet [13], into our framework. PFNet is not only suitable for simulating natural human activity in the form of a keypoint representation, but also allows for direct interaction with the model, which is a crucial requirement for real-time game engines. Note however, that our framework is also able to synthesize offline, i.e. recorded videos of a given behavior, which can be easily represented by a sequence of corresponding shape descriptors.

For the rendering of a person we train a generative network conditioned on pose representations obtained from the motion model and on latent variables representing the appearance of a person.

In the following, we first briefly describe PFNet. To be able to render the resulting motion, we need to introduce a projection between world- and view coordinates of the rendered person in the scene. Furthermore, the domain shift from keypoint representations used for training our generative rendering model to those returned by the previous step must be addressed. Finally we present our model for conditional image generation.

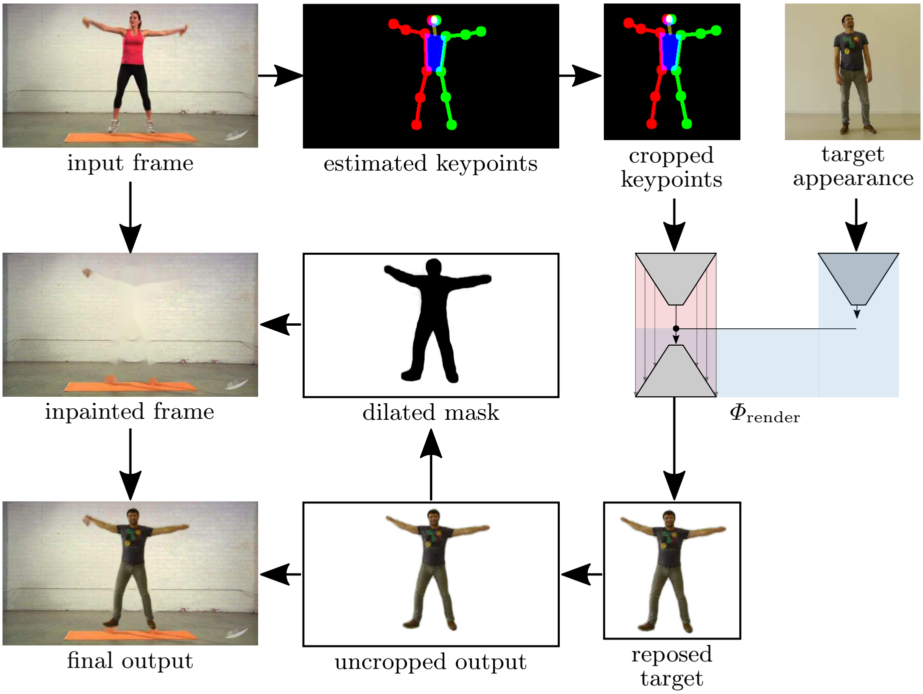

Fig. 3. Our pipeline for behavior cloning: Given an input frame and a target appearance, we first estimate keypoints of the input frame and a transformation to a bounding box. The resulting cropped keypoints are then rendered with the target appearance. We apply the inverse transformation to align the reposed target image with the original input frame. Before alpha blending this output with the input frame, we perform inpaininting to remove the original person.

3.1 Simulating Natural Human Behavior

When generating sequences of human poses, such as for animations or in computer games, one usually has a good idea of what kind of action should be the result. To generate highly realistic human action sequences so called Phase-Functioned Neural Networks (PFNN) [13] make use of the intrinsic periodicity of these motions. PFNNs such as that of [13] can be very shallow neural networks. Instead of learning a single set of model parameters these networks are trained to learn four sets of weights θ_i, which are interpolated given a phase p by means of a Catmul-Rom spline (c-spline) interpolation Θ:

$$\theta = \Theta\left(\theta_0, \theta_1, \theta_2, \theta_3\right) \ . \tag{1}$$

In this work a three layer fully connected neural network Φ_{PFN} with 512 hidden units and Exponential Linear unit activation functions (ELU) σ_{ELU} is employed, using these interpolated weights θ. At each time step t the network computes the joint locations of a human skeleton, in the following called stick man $\hat{y}^{3D}$, as well as the phase update Δp. Letting the network choose the rate of change of p can be seen as having the network choose the rhythm of the current motion pattern. To get a smooth update also the past and estimated future trajectory $\mathcal{T}$ at a total of 12 time steps is also given as an input. To control, which kind of behaviour is generated, the network additionally accepts a control input c, which allows a user to specify the desired motion pattern, such as walking or running and its orientation.

$$\hat{y}^{3D}_t, \Delta p_t = \Phi_{\mathrm{PFN}}\left(\hat{y}^{3D}_{t-1}, c_t, p_t, \mathcal{T}_{t-6:t+6}, \theta\right) \tag{2}$$

3.2 Domain adaption

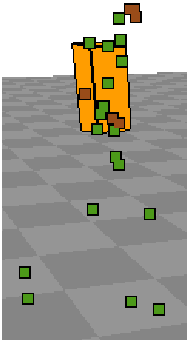

Fig. 4. Finding occluded keypoints using a simple polygon (shown in orange) and the view of the eyes, relative to the orientation of the camera. Occluded keypoints are shown in red, visible ones in green. Note that in our approach keypoints inside the orange polygon are marked as always visible.

Shown in Fig. 4 is an example of keypoints rendered as small squares given through $\hat{y}^{3D}$. Their 2d screen positions are now used to define the 2d pose input $\hat{y}^{2D}$ to generate the final render of the person. As our rendering model is trained given only the keypoints of visible body parts (see section 3.3) we filter the keypoints returned by PFNN in two ways: *(i)* Keypoints of arms and knees are marked as visible if the are not occluded by a polygon defined by the keypoints of the hip and each shoulder. It is visualised in orange in Fig. 4. Body keypoints are assumed to be always visible. Our experiments show that although being very simple this approximation is sufficient. *(ii)* Let $\mathbf{x}_{eye}$ be the vector describing the position of an eye keypoint, $\mathbf{x}_{nose}$ be the point in space describing the position of the nose and $\mathbf{x}_{cam}$ be the position of the camera. Each eye $i \in 0, 1$ is visible if

$$0 > \left(\frac{\mathbf{x}_{cam} - \mathbf{x}_{nose}}{\left\|\mathbf{x}_{cam} - \mathbf{x}_{nose}\right\|} \times (-1)^i \mathbf{e}_y \right) \cdot \frac{\mathbf{x}_{nose} - \mathbf{x}_{eye}^{(i)}}{\left\|\mathbf{x}_{nose} - \mathbf{x}_{eye}^{(i)}\right\|} \tag{3}$$

with:

$$i = \begin{cases} 0 & \text{if left eye} \\ 1 & \text{if right eye} \end{cases}, \tag{4}$$

where $\mathbf{e}_y$ is the unit vector in y or up direction, × is the cross product and · is the scalar product. See Fig. 5 for a visualization. The nose is marked visible if one or both eyes can be seen. Occluded keypoints are marked red in Fig. 4, visible ones green. Note that the skeleton looks away from the camera and the left arm is occluded by the body.

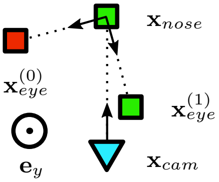

Fig. 5. The visibility of the eyes is determined by their orientation relative to the camera. See equation 3 for details. If one or both eyes are visible, the nose can be seen as well. In this visualization $\mathbf{e}_y$ points upwards, out of the paper plane towards the reader.

3.3 Rendering

Our goal is to learn the rendering of characters from natural images. To achieve this, we train a neural network to map normalized 2D pose configurations to natural images. With this network available, we then project the 3D keypoints obtained in the previous step to 2D coordinates according to the desired camera view. After normalization of this configuration, we can apply the network to obtain the rendering. Finally, the normalization transformation is undone to be able to blend the character into the scene at the correct position. The different steps are explained in more detail in the following.

Coordinate normalization In order to train the network, it would be very inefficient to predict images at different positions because the renderings should be translation invariant. Therefore, we use the 2D joint coordinates to define a region of interest which covers the joints and add 10% padding, to account for the character volume. This results in a transformation M_{coord} which can be used to transform points as well as images using bilinear resampling.

Image and Mask prediction For training, we assume that we have a large number of images of characters in different poses. As it will be important to provide images of a wide variety of poses, we utilize a network architecture which allows us to train a single network on multiple characters to increase the number of available poses.

For each training image of a character, we extract 2D joint positions and segmentation masks. Using large scale labeled data for these tasks such as [21], reliable estimation is possible. However, because we cannot predict positions of occluded joints, we must be careful to simulate occluded joints during test time as described in section 3.2.

Due to automatic estimation of joints and mask, we only require a large number of images of characters, which can be achieved efficiently by video recordings. In order to make use of training data obtained from multiple characters, our network must be able to disentangle the pose from the character's appearance, a task which has also been considered in [10]. Following these approaches, our network has two inputs, one for the joint positions and one for the character image. For preprocessing, the joint positions are converted into a stickman image to be able to utilize skip connections as in a U-Net architecture [28]. Furthermore, body parts are cropped from the character image to make sure that the network has to use the stickman image to infer joint locations. The training objective of the network is then given by reconstruction tasks on the original image as well as the segmentation mask. For the mask, we use a pixelwise $L1$ loss and for the images we use a perceptual loss which is highly effective as a differentiable metric for perceptual similiarity of images [38].

Merge with scene Finally, we use the inverse coordinate normalization M_{coord}^{-1} to transform the rendering and the mask back to their original screen coordinates. In order to integrate the character with the virtual environment, we use alpha blending between the rendered virtual environment and the character rendering. See also Fig. 2. In the case of behavior cloning from video, we first utilize the mask of the generated image to perform Image-Inpainting on the original frame. The full pipeline for this case is shown in Fig. 3.

4 Experiments

We evaluate our rendering framework qualitatively and quantitatively using dataset consisting of video sequences of three persons. The subjects were filmed performing various actions like walking, running, dancing, jumping and crouching. The recordings were done according to three settings: First a free setting, where actors were encouraged to perform a wide variety of movements without restrictions. For evaluation purposes, we also collected videos in a restricted setting, where the actors were restricted to standing still and performing only a walking motion. Filming was done using a Panasonic GH2 in full HD 1080p at 24 frames per second. Each individual is shown in approximately 10 000 frames. For training our conditional generative model, all frames are annotated with keypoints using the openpose [6] library. Additionally a mask covering the person in each frame is calculated using the deeplab [8] toolbox. Note that filming could also be done with a cellphone camera or something similar, making this approach feasible for a wide range of audiences.

4.1 Qualitative Results in Virtual Environment

For qualitative evaluation of the rendering capabilities of our framework in a virtual, interactive environment, we adapted the testing API of PFNet[13]. Our rendering model is trained on all training images of our 3 subjects. While inference we extract the simulated keypoints of the API, project it from 3D world- to 2D view coordinates while accounting for the domain shift and condition the appearance rendering on the resulting output. Fig. 6 and 7 show our renderings for two different simulated walking scenarios, given the shape conditioning and different person appearances. Further, in Fig. 8 and 9 we demonstrate the need for self-occlusion handling. Additionally Fig. 10 shows an ablation experiment, where the same behavior as in Fig. 9 is rendered using nearest neighbor frames from the training set. This experiment clearly demonstrates the ability of our model to interpolate between the training images for generating smooth transitions in appearance while simulating a given behavior. A video with examples can be found at https://compvis.github.io/hbugen2018/.

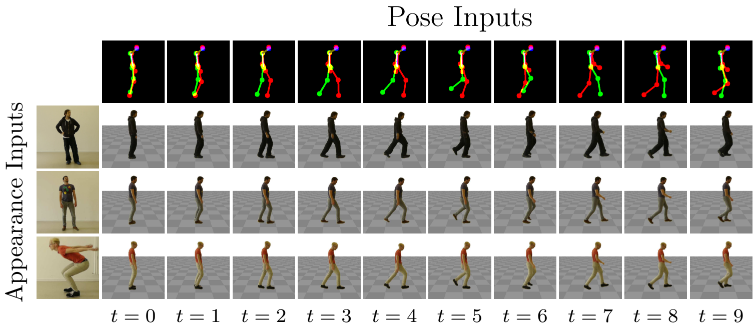

Fig. 6. Walking sequence perpendicular to the viewer. Note how the model consistently manages to generate the same pose for each appearance in each frame.

4.2 Qualitative Results on Video Data

We now show additional qualitative results by simulating different behaviors as shown in video sequences. We trained our model using the full training sets of our test subjects and applied it on keypoint trajectories extracted from the PennAction dataset[39]. It contains 2326 video sequences of 15 different sports

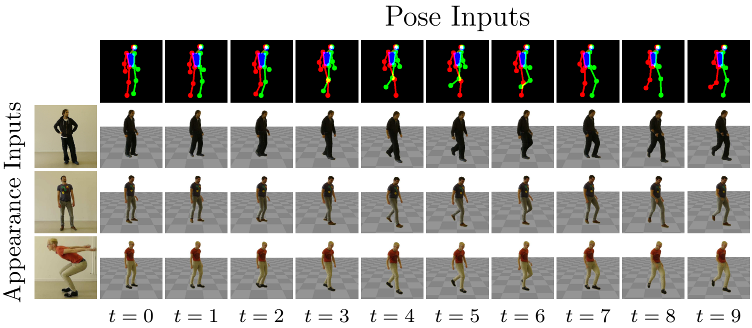

Fig. 7. Walking sequence towards the viewer at an angle. Note that our model can generalize not only to different poses, but also to different perspectives.

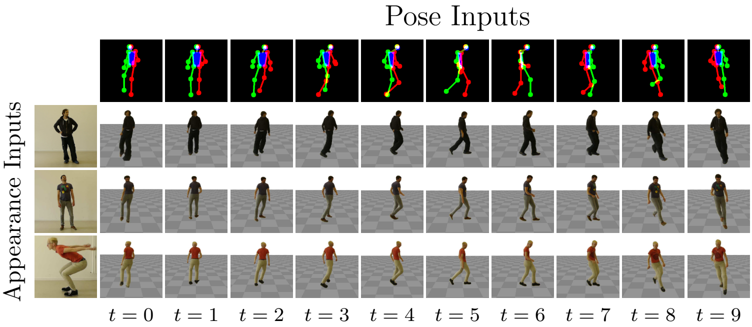

Fig. 8. Walking in circles *without* occlusion modeling. Note how at time steps $t = 0$ to 6 there appear eyes on the back of the heads of the three characters. Compare this to Fig. 9, where occlusion modeling is applied.

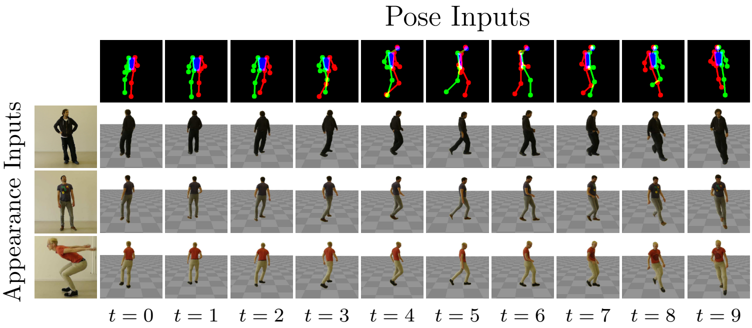

Fig. 9. Walking in circles *with* occlusion modeling. Note how at the first 6 time steps there are no eyes at the back of the heads, as opposed to Fig. 8.

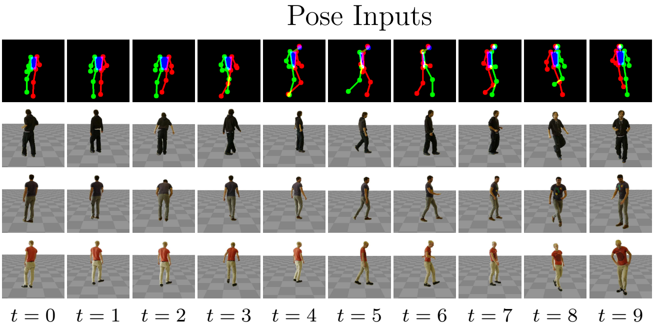

Fig. 10. Walking in circles with occlusion modeling. Here we show the nearest neighbors to each pose. Note how the images from time step to time step are quite different or just stay the same. There is also no consistency of pose with varying appearance as one can see, when comparing each image at a single time step.

categories. The dataset exhibits unconstrained activities including complex articulations and self-occlusion. Fig. 11 shows example renderings randomly selected from different activities simulated by different persons. A video with examples can be found at https://compvis.github.io/hbugen2018/. Further in Fig. 12 - 15 re-enactments of exemplary target behaviors are illustrated by temporal sampling the source videos. Conditioned on the estimated pose from the individual frames, we infer a new appearance and project the rendering back into the source frame. Thus we are able to simulate the given activities by any person of our choice.

4.3 Quantitative Evaluation of Pose Generalization

training setting	Standing			Walking			Full		
	query	augmented	NN	query	augmented	NN	query	augmented	NN
SSIM	0.841	0.858	0.771	0.848	0.863	0.782	0.908	0.914	0.794

Table 1. Structural similarity scores (SSIM) for different training settings. 'query' refers to test person data only and 'augmented' refers to additional data augmentation. 'NN' denotes the nearest neighbor retrieval results.

Let us now quantitatively evaluate the ability of our model to generalize to unseen postures. For this experiment we train our model on three different training subsets of varying variance in body articulation featuring only a single person: *(i)* Only images showing the person while standing with relaxed arms, *(ii)* Only images the showing a test person walking up and down and *(iii)* the

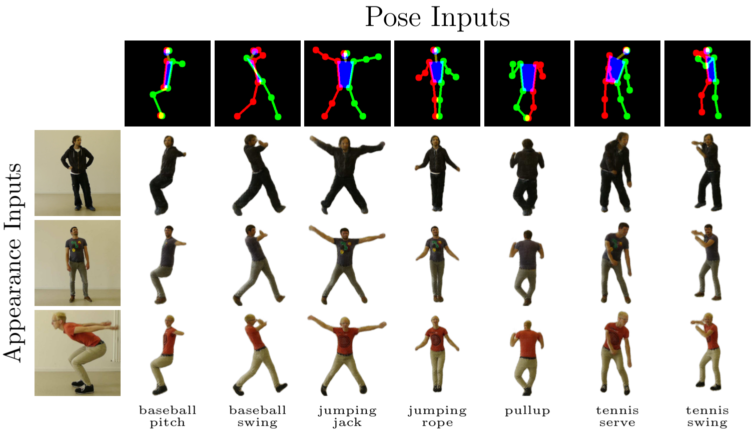

Fig. 11. Using randomly sampled poses from the PennAction dataset, our model is able to generate realistic renderings conditioned on different appearances.

person's full training set. Moreover, we also train models for each of these settings with additional data augmentation by adding the full training sets of the remaining test subjects. We then compute the mean structural similarity score (SSIM)[36] between groundtruth test images and renderings of our model based on their extracted postures. As a baseline we use nearest neighbor retrievals from the different training sets also based on the extracted keypoints. Table 1 summarizes the results. As one can see, with increasing variability of the training poses also the quality of the renderings improves. Moreover, data augmentation in form of additional images persons helps our model to interpolate between the training poses of the actual test subject and thus improves its generalization ability. Note that on average our model outperforms the baseline by 9.5%, which proves that our model actually understands the mapping between shape and appearance.

5 Conclusion

In this work we presented an approach towards a holistic learning framework for rendering human behavior. Both rendering the appearance of a person and its simulating natural movements while performing a given behavior are represented by deep neural networks, which can be trained from easily available RBG video data. Our model utilizes a conditional generative model to learn a mapping between abstract pose representation and the appearance of a person. Using this model, we are able to simulate any kind of behavior conditioned on the appearance of a given test subject while either directly controlling the behavior in a virtual environment or reenacting recorded video sequences.

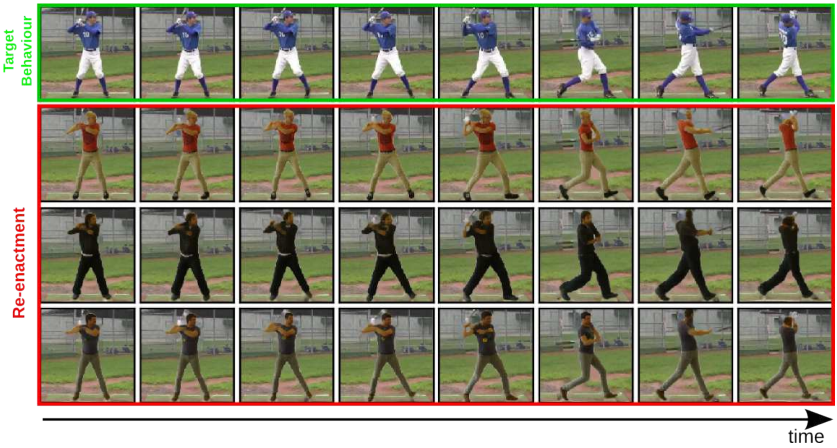

Fig. 12. Re-enactment of ’Baseball swing’. Green illustrates the target behavior and red its simulation based on different appearances. Frames are uniformly sampled in time.

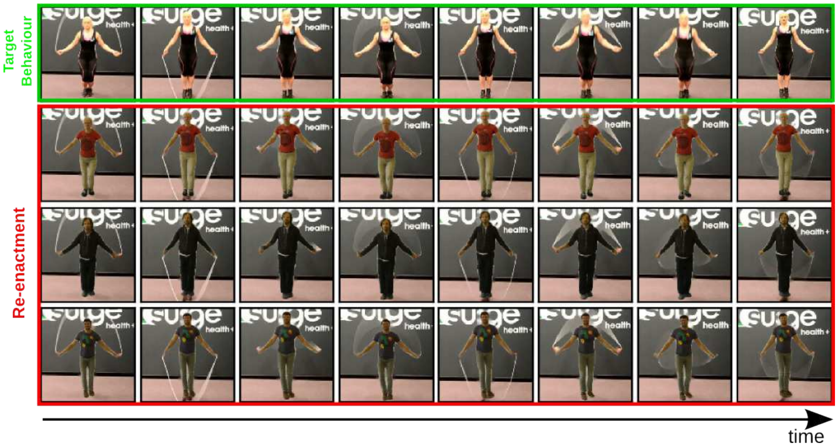

Fig. 13. Re-enactment of ’jumping rope’. Green illustrates the target behavior and red its simulation based on different appearances. Frames are uniformly sampled in time.

This work has been supported in part by DFG project 371923335 and a hardware donation from NVIDIA.

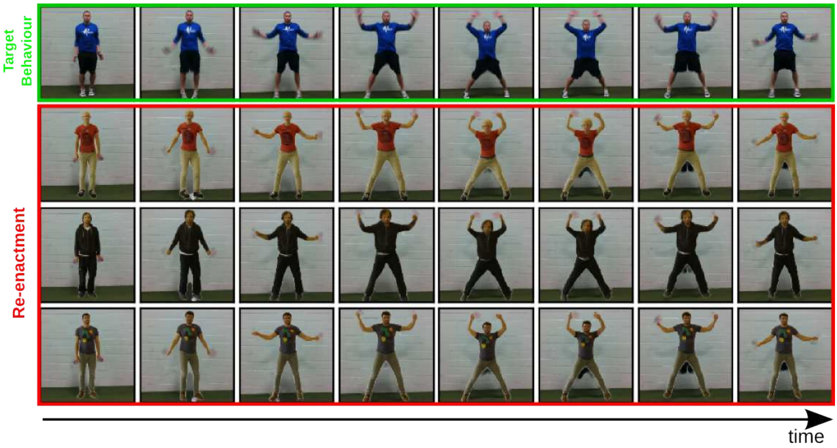

Fig. 14. Re-enactment of ’jumping jack’. Green illustrates the target behavior and red its simulation based on different appearances. Frames are uniformly sampled in time.

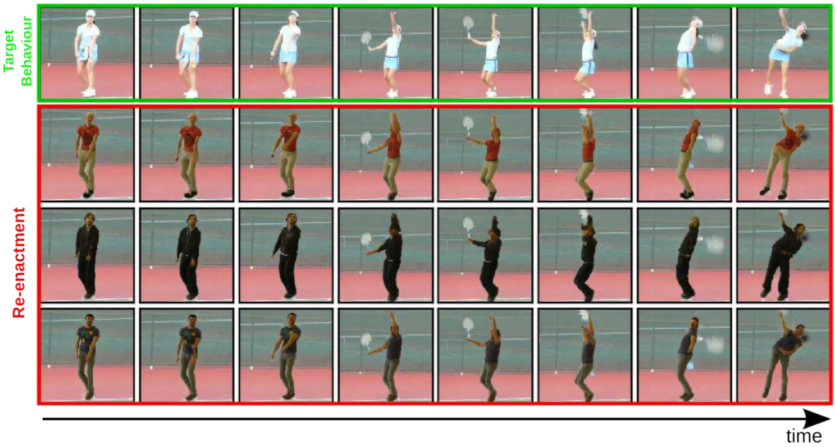

Fig. 15. Re-enactment of ’tennis serve’. Green illustrates the target behavior and red its simulation based on different appearances. Frames are uniformly sampled in time.

References

[1] B. Allain, J.-S. Franco, and E. Boyer. An efficient volumetric framework for shape tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. *ACM Trans. Graph.*, 2005.

[4] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag. Synthesizing images of humans in unseen poses. *arXiv preprint arXiv:1804.07739*, 2018.

[5] C. C., B. E., and I. S. Probabilistic deformable surface tracking from multiple videos. In *European Conference on Computer Vision (ECCV)*, 2010.

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[7] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. In *ACM SIGGRAPH 2003 Papers*, 2003.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *ArXiv e-prints*, June 2016.

[9] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 2008.

[10] P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. *arXiv preprint arXiv:1804.04694*, 2018.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 2014.

[12] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A Statistical Model of Human Pose and Body Shape. *Computer Graphics Forum*, 2009.

[13] D. Holden, T. Komura, and J. Saito. Phase-functioned neural networks for character control. *ACM Trans. Graph.*, 2017.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016.

[15] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[16] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[18] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] C. Lassner, G. Pons-Moll, and P. V. Gehler. A generative model of people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 6, 2017.

[20] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. Graph.*, 2013.

[21] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft COCO: Common Objects in Context. *ArXiv e-prints*, May 2014.

[22] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2015.

[23] L. Ma, Q. Sun, S. Georgoulis, L. V. Gool, B. Schiele, and M. Fritz. Disentangled person image generation. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[24] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. 2017.

[25] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

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

[27] A. Popa, M. Zanfir, and C. Sminchisescu. Deep multitask architecture for integrated 2d and 3d human sensing. In *CVPR*, 2017.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] A. Shapiro, A. Feng, R. Wang, H. Li, M. Bolas, G. Medioni, and E. Suma. Rapid avatar capture and simulation using commodity depth sensors. *Computer Animation and Virtual Worlds*, 2014.

[30] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe. Deformable gans for pose-based human image generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] D. Tomè, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *CVPR*, 2017.

[32] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, 2017.

[33] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016.

[35] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems*. 2016.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 2004.

[37] M. Zeng, J. Zheng, X. Cheng, and X. Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *CVPR*, 2013.

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint*, 2018.

[39] W. Zhang, M. Zhu, and K. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *International Conference on Computer Vision (ICCV)*, 2013.

[40] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[41] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[43] S. Zuffi and M. J. Black. The stitched puppet: A graphical model of 3D human shape and pose. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, 2015.