This CVPR 2021 workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Differentiable Rendering-based Pose-Conditioned Human Image Generation

Yusuke Horiuchi
Waseda University
y.horiuchi@suou.waseda.jp

Edgar Simo-Serra
Waseda University
ess@waseda.jp

Satoshi Iizuka
University of Tsukuba
iizuka@cs.tsukuba.ac.jp

Hiroshi Ishikawa
Waseda University
hfs@waseda.jp

Abstract

Conditional human image generation, or generation of human images with specified pose based on one or more reference images, is an inherently ill-defined problem, as there can be multiple plausible appearance for parts that are occluded in the reference. Using multiple images can mitigate this problem while boosting the performance. In this work, we introduce a differentiable vertex and edge renderer for incorporating the pose information to realize human image generation conditioned on multiple reference images. The differentiable renderer has parameters that can be jointly optimized with other parts of the system to obtain better results by learning more meaningful shape representation of human pose. We evaluate our method on the Market-1501 and DeepFashion datasets and comparison with existing approaches validates the effectiveness of our approach.

1. Introduction

Generation of novel human images from a reference image has many different applications such as virtual try-on, virtual/augmented reality, art, video manipulation, etc. However, there can be large ambiguities in the output from occlusions in the reference image. In order to minimize the uncertainty and to focus on real world applications, here we focus on conditioning the image generation on multiple reference images.

Most existing approaches for conditional human image generation from another pose rely on heatmaps that encode the pose information [5, 6, 7]. In general, this heatmap is generated from sparse points by fitting Gaussians or other simple structures, often significantly impacting the performance. In this work, we propose to use a differentiable rendering engine that can learn the optimal way to generate the heatmaps from the data itself, instead of using the various heuristics.

We build our approach upon Multiple-Source DeformableGAN (MS-DefGAN) [7], which is applicable to multiple reference images, and we modify it to use a differentiable heatmap renderer that takes the pose inputs. Unlike previous approaches, the input heatmap represent the pose not only through keypoints, but also through edges, so that better conditional dependence can be learned from the rendering of heatmaps such as arms and legs as well as elbows and knees. This richer representation is able to encode more details of the human body. The differentiable renderer is parametrized and designed so it can learn fundamental parameters of the human body shape through backpropagation, improving the accuracy of the human image generation using multiple reference images. An overview of our approach is shown in Fig. 1.

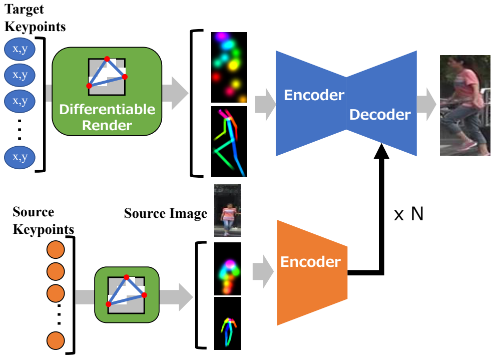

Figure 1. Overview of our method. From the input pose keypoints, our differentiable renderer generates a pose condition heatmaps that represent both keypoints and edges.

We evaluate our approach on the Market-1501 and Deepfashion datasets and demonstrate the effectiveness of our approach.

2. Related Work

Conditional human image generation, otherwise known as the pose conversion, is a task to generate an image of a person in given reference images but with a specified alternative pose [5]. Existing approaches differ in how the pose is specified, and are roughly divided into ones that use an image of another person to specify the pose and the ones that use a 2D skeleton. In the former, procuring the image of a person with the desired pose can be cumbersome or even impossible. Also, the generated image tend to be influenced by the pose-specifying image's appearance, from which we only want the pose. On the other hand, using the skeleton to specify the pose makes the task more difficult and, as a result, the quality of the generated image can suffer, though it is easier to use and more convenient from the user's point of view. There also exist methods that utilize optical flow, which require moving images as the input, as well as ones that perform texture mapping on a 3D model of the person prepared in advance. In this paper, we choose the more difficult approach, which is the use of skeleton input, for versatility's sake.

DefGAN [6] is known as a skeleton-based method. It realizes high-performance pose conversion without using optical flow or 3D model through cut-and-paste of feature maps to transform the original pose to the target pose, through a clever use of skip connection in the U-Net structure. Its variant SA-DefGAN [3] realizes higher-quality transformation that considers the entire image by the use of Global Self-Attention in DefGAN. However, since the computation cost of SA-DefGAN increases as the square of the number of pixels, its use generalizes poorly to higher-resolution images required in practical use. MS-DefGAN [7] is a more practical extension of DefGAN. It can use single or multiple reference images. Using multiple images can help reduce ambiguities and increase performance. However, MS-DefGAN has been reported to have a lower SSIM score than VUnet [2, 7], which uses Variational Auto Encoder to generate looks and poses. The VUnet, in turn, has the disadvantage of not generating very diverse outputs relative to GAN-based methods, nor is it easily extended to the multi-source context, as it largely abandons spatial information, making it hard to correlate multiple input images.

In this paper, we propose a higher-performance conditional human image generation framework based on MS-DefGAN, in which we emphasize the importance of learning to generate the shape more precisely. Our proposed method can help in any of the DefGAN variations to provide a better shape-generation ability. In existing research, the pose is given to the image-generating network as a pseudo-image, or a heatmap, depicting the shape of desired human pose. This shape generation is often unsuccessful, as is the case of [6], leading to the preference of texture cut-and-paste. It has been expressed in terms of dots, the fixed size of which is determined by a hyperparameter search. Our method represents the human pose as dots for important points such as the hand, elbows, knees, and feet, as well as the edges that connect these points. It also automatically learns the size of the dots and the lengths and the thicknesses of the edges to create an appropriate pose heatmap. Since this method only learns a small number of parameters, it can be used in any of the above methods without adding much overhead after learning.

3. Our Approach

Our method is based on MS-DefGAN [7], extended to have a pose keypoint rendering module (Vertex Renderer) and pose skeleton rendering module (Edge Renderer).

The pose information P for conditioning is a tuple of a fixed number of two-dimensional points:

$$P = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_k), \tag{1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^2$. In the experiments we set $k = 18$.

Based on this pose information P, a vertex heatmap $H_{V_i} \in \mathbb{R}^{H \times W}$ is rendered for each vertex V_i, and an edge heatmap $H_{E_i} \in \mathbb{R}^{H \times W}$ is rendered for each edge E_i, where H and W are the height and the width of the output image, respectively.

3.1. Vertex Renderer

In the DefGAN [6], each keypoint heatmap is given as:

$$H_{V_i}(p) = \exp\left(-\left\|p - \boldsymbol{p}_i\right\| / \sigma^2\right) \tag{2}$$

for each pixel coordinate p. The conditioning pose information is fed to the network as concatenation of the heat maps in the channel direction for each pixel/node:

$$\begin{aligned} H_V &= \mathrm{VertexRender}(P) && (3) \\ &= \mathrm{Concat}(H_{V_1}, H_{V_2}, \ldots, H_{V_k}) && (4) \end{aligned}$$

In DefGAN, $\sigma = 6.0$ was set as a hyperparameter from the results obtained by cross-validation, and pose information was calculated using this. However, setting hyperparameters is very troublesome, and the optimal value can change depending on the size of the image and the nature of the data set. Therefore, in our approach, the value of σ is learned for each vertex. This is implemented by preparing and calculating a tensor with a sequence of numbers. Note that it will be 0 if the keypoint is not detected.

3.2. Edge Renderer

There is predetermined keypoint connection information $E = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{19})$, where each element $\boldsymbol{e}_i = (j_i, k_i)$ is a pair of vertex indices, meaning $\boldsymbol{p}_{j_i}$ and $\boldsymbol{p}_{k_i}$ are the endpoints of the edge. The edge is extended by the learnable parameters α_{s_i} and α_{e_i}, and the end points of the edge are given as follows:

$$\begin{aligned} \boldsymbol{p}_{\mathrm{start}_i} &= \boldsymbol{p}_{j_i} + \alpha_{s_i}\left(\boldsymbol{p}_{k_i} - \boldsymbol{p}_{j_i}\right) && (5) \\ \boldsymbol{p}_{\mathrm{end}_i} &= \boldsymbol{p}_{k_i} + \alpha_{e_i}\left(\boldsymbol{p}_{k_i} - \boldsymbol{p}_{j_i}\right). && (6) \end{aligned}$$

Using the vertices, we determine the vector in the edge and the orthogonal directions. The norm of the vector is ad-

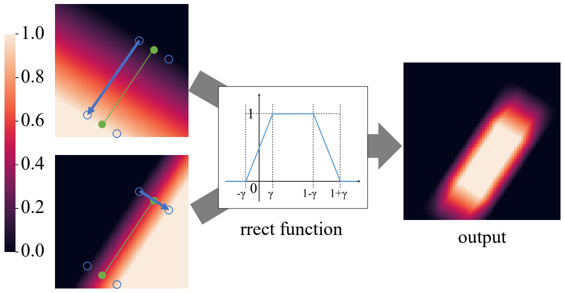

Figure 2. Visualization of the Edge Rendering Algorithm. The blue arrow indicates the direction of the vector used to calculate the inner products. The inner product of the vector from the start point of the arrow to each position and the blue arrow is shown in changing colors. Black pixels represent values of 0 or less, and white pixels represent values of 1 or more. In this method, the part containing the value from 0 to 1 is drawn.

justed so that it is the reciprocal of the original length.

$$\boldsymbol{v}_{\text{straight}} = \frac{\boldsymbol{p}_{\text{end}} - \boldsymbol{p}_{\text{start}}}{\left\|\boldsymbol{p}_{\text{end}} - \boldsymbol{p}_{\text{start}}\right\|^2 + \epsilon} \tag{7}$$

$$\boldsymbol{v}'_{\text{normal}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \boldsymbol{v}_{\text{straight}} \tag{8}$$

$$\boldsymbol{v}_{\text{normal}} = \beta \, \frac{\boldsymbol{v}'_{\text{normal}}}{\left\|\boldsymbol{v}'_{\text{normal}}\right\|^2 + \epsilon} \tag{9}$$

A rectangular heat map is created as the inner products of each positions and these vectors. We then employ a ramped rectangular function (rrect) [4] to allow back-propagation for the sloped part:

$$\begin{aligned} H_{E_i}(p) = & \operatorname{rrect}\left((p - \boldsymbol{p}_{\text{start}}) \cdot \boldsymbol{v}_{\text{straight}}\right) \\ & \cdot \operatorname{rrect}\left((p - \boldsymbol{p}_{\text{start}}) \cdot \boldsymbol{v}_{\text{normal}} + 0.5\right). \end{aligned} \tag{10}$$

The complete edge heatmap tensor is then created by concatenating the individual heatmaps as:

$$\begin{aligned} H_E &= \text{EdgeRenderer}(P) && (11) \\ &= \text{Concat}(H_{E_1}, H_{E_2}, \ldots, H_{E_k}) && (12) \end{aligned}$$

The rrect function is defined as follows:

$$\operatorname{rrect}(x) = \begin{cases} 0 & (x < -\gamma) \\ \frac{1}{2\gamma}(x + \gamma) & (-\gamma \leq x < \gamma) \\ 1 & (\gamma \leq x < 1-\gamma) \\ -\frac{1}{2\gamma}(x-1-\gamma) & (1-\gamma \leq x < 1+\gamma) \\ 0 & (1+\gamma \leq x) \end{cases} \tag{13}$$

$$\begin{aligned} &= \min\left(\max\left(\frac{1+\gamma-x}{2\gamma}, 0\right), 1\right) \\ &\quad + \min\left(\max\left(\frac{\gamma+x}{2\gamma}, 0\right), 1\right) - 1 \end{aligned} \tag{14}$$

Table 1. Comparison with other pose transfer methods on the Market-1501 for different number of reference images n. Since [2, 6] are methods that supports only a single source, these are evaluated only on input 1. MS-DefGAN [7] and our methods support multiple sources, these are evaluated even if there are multiple inputs. mSSIM and mIS indicate masked SSIM and masked Inception Score, respectively.

Model	n	SSIM	IS	mSSIM	mIS
VUnet [2]	1	0.312	3.283	0.862	2.544
DefGAN [6]	1	0.225	2.994	0.828	2.745
MS-DefGAN [7]	1	0.318	3.256	**0.863**	**2.593**
Ours	1	**0.324**	**3.316**	0.860	2.538
MS-DefGAN [7]	3	0.362	3.173	**0.877**	**2.541**
Ours	3	**0.364**	**3.220**	0.873	2.494
MS-DefGAN [7]	10	0.384	3.082	**0.885**	**2.510**
Ours	10	**0.384**	**3.083**	0.881	2.471

Our EdgeRenderer is thus able to automatically adjust the length in the edge direction and the edge thickness in the perpendicular direction. A visualization of this method is shown in Fig. 2.

The edge elongation factor α's and the edge thickening factor β's are learned with back-propagation. The slope of the edge γ is fixed to 0.2 in this study.

4. Experiments

4.1. Datasets

We compare our method and pre-existing methods (VUnet, DefGAN, and MS-DefGAN) using the Market-1501 and DeepFashion datasets. The poses of the persons in the images in the datasets are estimated and annotated by using OpenPose [1] . We create input-output pairs from images of the same person. For the Market-1501 dataset we only use images of people with at least 13 instances, while for the DeepFashion dataset we only use images of people with at least 4 instances.

As a result, we created 215,750 training, 23,913 validation, and 23,491 evaluation pairs from the Market-1501 dataset, and 26,014 training, 2,900 validation, and 6,708 evaluation pairs from the DeepFashion dataset. Note that VUnet is a VAE, so it is trained on a single image rather than paired training data. VUnet is trained with 8,803 Market1501 mages and 20,902 DeepFashion images.

4.2. Training

The structure of the neural network and the loss function follow those of MS-DefGAN. Note that the MS-DefGAN [7] model is pre-trained with two reference images and then re-trained before the testing using the same number of input images as the test data. Here, for fairness,

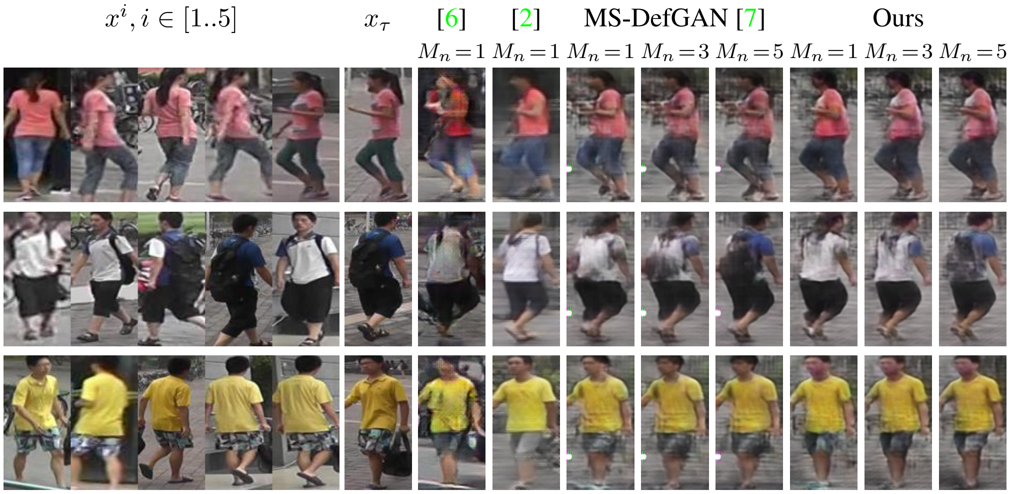

Figure 3. A qualitative comparison on the Market-1501 dataset. The first column shows the source images. [6] and [2] use only the first source image. The target poses are given by the ground truth images in column 2. In column 4, we show the results obtained by our model from increasing numbers (M_n) of source images. The source from the first column are added while increasing M_n from left to right. We can see that the overall shapes of the persons look better in our results, such as the pants.

we train the models from the beginning to the end with a fixed number of inputs and evaluate using the same model without re-training. The Market-1501 dataset model is trained with 12 input images, and the DeepFashion dataset model is trained with 3 inputs.

The training is terminated when the highest SSIM score for the validation dataset is reached. The SSIM score is calculated with the window size of 11 and in 8bit color.

4.3. Quantitative Comparison

We quantitatively evaluate the output image quality of the models trained as in 4.2 in each input image number setting.

As evaluation metrics, SSIM, masked SSIM, Inception Score (IS), and masked Inception Score are used. The result is shown in Tables 1 and 2. The higher SSIM represents the higher degree of shape matching. The Inception Score evaluates the diversity of generated images. VUnet [2], with its use of VAE, significantly sacrifices the IS to enhance the SSIM. Since our method uses the renderer to generate the shape, it can enhance the SSIM without sacrificing the IS. Our results have the best SSIM, albeit slightly.

4.4. Qualitative Comparison

The outputs by the methods can be compared in Fig. 3 for the Market-1501 dataset and in Fig. 4 for the DeepFashion dataset. We can see that the overall shapes of the persons look better in our results.

5. Conclusion

We have presented an approach for pose-conditioned human image generation based on using a differentiable rendering engine. Our approach encodes the human pose more accurately by representing it with both vertices and edges. Furthermore, instead of relying on heuristics to generate the pose heatmaps, our approach is able to determine the generation parameters directly from the training data through back-propagation. Experiments on the Market-1501 and DeepFashion datasets corroborate the effectiveness of our approach.

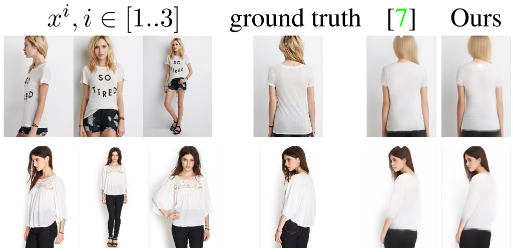

Figure 4. Qualitative results on DeepFashion Dataset.

Table 2. Quantitative comparison on DeepFashion dataset.

Model	n	SSIM	IS
VUnet [2]	1	0.763	3.025
DefGAN [6]	1	0.695	**3.139**
MS-DefGAN [7]	1	0.764	2.916
Ours	1	**0.767**	2.770
MS-DefGAN [7]	3	0.779	**2.811**
Ours	3	**0.779**	2.670

6. Acknowledgements

This work was partially supported by JSPS Grant-in-Aid for Scientific Research (A) grant number 20H00615.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7291–7299, 2017. 3

[2] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 3, 4

[3] Yusuke Horiuchi, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Spectral normalization and relativistic adversarial training for conditional pose generation with self-attention. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–5, 2019. 2

[4] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3907–3916, 2018. 3

[5] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Adv. Neural Inform. Process. Syst.*, pages 405–415, 2017. 1

[6] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 2, 3, 4

[7] Lathuilière Stéphane, Sangineto Enver, Siarohin Aliaksandr, and Sebe Nicu. Attention-based fusion for multi-source human image generation. In *IEEE Winter Conf. Applic. of Comput. Vis.*, pages 428–437, 2020. 1, 2, 3, 4