Dense Intrinsic Appearance Flow for Human Pose Transfer

Yining Li[1] Chen Huang[2] Chen Change Loy[3]
[1]CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
[2]Robotics Institute, Carnegie Mellon University
[3]School of Computer Science and Engineering, Nanyang Technological University
ly015@ie.cuhk.edu.hk chenh2@andrew.cmu.edu ccloy@ntu.edu.sg

Abstract

We present a novel approach for the task of human pose transfer, which aims at synthesizing a new image of a person from an input image of that person and a target pose. Unlike existing methods, we propose to estimate dense and intrinsic 3D appearance flow to better guide the transfer of pixels between poses. In particular, we wish to generate the 3D flow from just the reference and target poses. Training a network for this purpose is non-trivial, especially when the annotations for 3D appearance flow are scarce by nature. We address this problem through a flow synthesis stage. This is achieved by fitting a 3D model to the given pose pair and project them back to the 2D plane to compute the dense appearance flow for training. The synthesized ground-truths are then used to train a feedforward network for efficient mapping from the input and target skeleton poses to the 3D appearance flow. With the appearance flow, we perform feature warping on the input image and generate a photorealistic image of the target pose. Extensive results on DeepFashion and Market-1501 datasets demonstrate the effectiveness of our approach over existing methods. Our code is available at http://mmlab.ie.cuhk.edu.hk/projects/pose-transfer/

1. Introduction

The ability to predict what an object will look like from a new viewpoint is fundamental to intelligence. Human pose transfer [26] is an important instantiation of such view synthesis task. Given a single view/pose of one person, the goal is to synthesize an image of that person in arbitrary poses. This task is of great value to a wide range of applications in computer vision and graphics. Examples include video synthesis and editing and data augmentation for problems like person re-identification where it is hard to acquire enough same-person images from different cameras.

Despite the rapid progress in deep generative models like Generative Adversarial Networks (GAN) [6] and Variational Auto Encoders (VAE) [17], human image generation between poses is still exceedingly difficult. The main challenge is to model the large variations in 2D appearance due to the change in 3D pose. This is further compounded by human body self-occlusion that induces ambiguities in inferring unobserved pixels for the target pose. In general, successful human pose transfer requires a good representation or disentangling of human pose and appearance, which is non-trivial to learn from data. The ability to infer invisible parts is also necessary. Moreover, the image visual quality largely depends on whether the high frequency details can be preserved, *e.g.* in cloth or face regions.

Most existing methods for human pose transfer [1, 5, 18, 23, 24, 27, 28, 47] employ an encoder-decoder architecture to learn the appearance transformation from an input image, guided by the input and target 2D pose encoded with some keypoints of the human-body joints. However, such keypoint-based representation is only able to capture rough spatial deformations, but not fine-grained ones. As a result, distortions or unrealistic details are often produced, especially in the presence of large pose change with non-rigid body deformations. Recent advances either decompose the overall deformation by a set of local affine transformations [34], or use a more detailed pose representation than the keypoint-based one. The latter is to enable 'dense appearance flow' computation that more accurately specifies how to move pixels from the input pose. Neverova *et al*. [26] showed that the surface-based pose representation via DensePose [7] serves as a better alternative. Zanfir *et al*. [44] turned to fit a 3D model to both input and target images, and then perform appearance transfer between the corresponding vertices. The resulting appearance flow with 3D geometry supervision is more ideal, but the 3D model fitting would incur too much burden at inference time.

In this paper, we propose a novel approach to human pose transfer that integrates implicit reasoning about 3D geometry from 2D representations only. This allows us to share the benefits of using 3D geometry for accurate pose transfer but at much faster speed. Our key idea is to recover

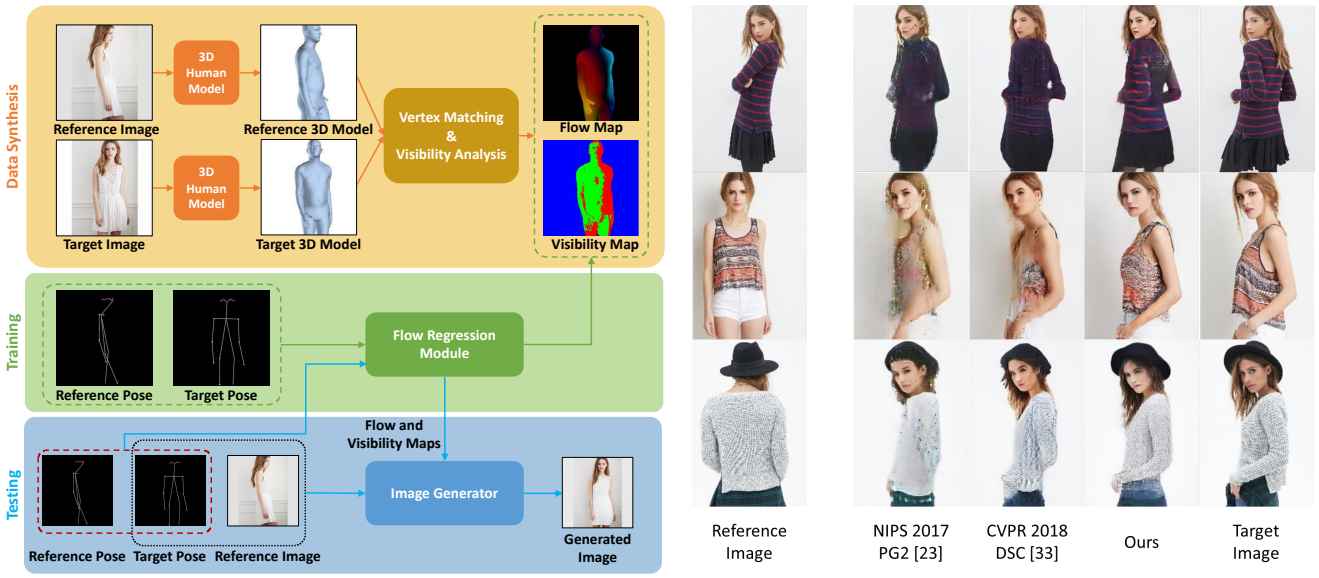

Figure 1: The proposed human pose transfer method with dense intrinsic 3D appearance flow generates higher quality images in comparison to baselines. (**Left**) The core of our method is a flow regression module (the green box) that can transform the reference and target poses into a 3D appearance flow map and a visibility map.

from training image pairs (along with their pose keypoints) the underlying 3D models, which when projected back to 2D image plane can provide the ground-truth appearance flow for us to learn from. Such *dense and intrinsic appearance flow* implicitly encodes the 3D structures of human body. Then we train an appearance flow generation module, represented by the traditional feedforward network, which directly regresses from a pair of 2D poses to the corresponding appearance flow. This module helps us to bypass the expensive 3D model fitting at test time, and predict the intrinsic pixel-wise correspondence pretty fast without requiring explicit access to 3D geometry.

Figure 1 (left) illustrates our overall image generation framework. Given a reference image (and its pose) and the target pose, we first use a variant of U-Net [29] to encode the image and target pose respectively. Then our appearance flow module generates a 3D flow map from the pose pair, and further generates a visibility map to account for the missing pixels in the target pose due to self-occlusions. The visibility map proves necessary for our network to synthesize missing pixels at the correct locations. To render the final image in target pose, the encoded image features are first warped through the generated flow map, and then passed to a gating module guided by the visibility map. Finally, our pose decoder concatenates such processed image features to generate the image. Our U-Net-type image generator and appearance flow module are trained end-to-end so as to optimize a combination of reconstruction, adversarial and perceptual losses. Our approach is able to generate high quality images on DeepFashion [20] and Market-1501 [48] datasets, showing consistent improvements over existing image generators based on keypoints or other pose representations. Our method also achieves compelling quantitative results.

The main contributions of this paper can be summarized as follows:

- A feedforward appearance flow generation module is proposed to efficiently encode the dense and intrinsic correspondences in 3D space for human pose transfer.
- An end-to-end image generation framework is learned to move pixels with the appearance flow map and handle self-occlusions with a visibility map.
- State-of-the-art performance and high quality images are produced on DeepFashion dataset.

2. Related Work

Deep generative image models. Recent years have seen a breakthrough of deep generative methods for image generation, using Generative Adversarial Networks (GAN) [6], Variational Autoencoder (VAE) [17] and so on. Among these, GAN has drawn a great attention due to its capability of generating realistic images. Follow-up works make GANs conditional, generating images based on extra inputs like class labels [25], natural language descriptions [33, 45, 46] or images from another domain [12] that leads to an image-to-image domain transfer framework. Adversarial learning has also shown its effectiveness in many other tasks like image super-resolution [19, 39, 40] and texture generation [42].

Human pose transfer. Generating human-centric images is an important sub-area of image synthesis. Example tasks range from generating full human body in clothing [18] to generating human action sequences [3]. Ma *et al*. [23] are the first ones to approach the task of human pose transfer, which aims to generate a person image in a target pose if a reference image of that person is given beforehand. The pose comprised of 18 keypoints, is represented as a 18-channel keypoint heatmap. Then it is concatenated with the reference image and fed into a two-stage CNN for adversarial training. Zhao *et al*. [47] adopted a similar coarse-to-fine approach to generate new images, but conditioned on the target view rather than target pose with multiple keypoints. To better handle the non-rigid body deformation in large pose transfer, Siarohin *et al*. [34] proposed Deformable GAN to decompose the overall deformation by a set of local affine transformations. Another line of works [5, 24, 27, 28] focus on disentangling human appearance and pose with weak supervision. With only single image rather than a pair as input, these methods try to distill appearance information in a separate embedding, sometimes with the help of cycle-consistent penalty [27].

Geometry-based pose transfer. Some recent works integrate geometric constraints of human body to improve pose transfer. Neverova *et al*. [26] proposed a surface-based pose representation on top of DensePose [7]. This allows to map the body pixels to a meaningful UV-coordinate space, where surface interpolation and inpainting can happen before warping back to the image space. Zanfir [44] on the other hand, proposed to leverage 3D human model to explicitly capture the body deformations. Specifically, they fit a 3D human model [21] to both source and target images using the method in [43], where a human body is represented by 6890 surface vertices. Then the pixels on overlapping vertices are directly transfered to the target image, while the invisible vertices in source image are hallucinated using a neural network. The main drawback of this work is that 3D model fitting is computationally expensive and is not always accurate. Our method avoids the costly 3D model fitting at test time, and instead learns to predict the 2D appearance flow map and visibility map defined by 3D correspondences in order to guide pixel transfer. This enables implicit reasoning about 3D geometry without requiring access to it.

Appearance flow for view synthesis. Optical flow [9] provides dense pixel-to-pixel correspondence between two images, and has been proved useful in tasks like action recognition in video [35]. Appearance flow [49] also specifies dense correspondence often between images with different view-points, which is closer to our setting. However, previous works mainly estimate appearance flow from simple view transformations (*e.g*., a global rotation) or rigid objects (*e.g*., a car). Whereas our appearance flow module deals with the articulated human body with arbitrary pose transformation.

3. Methodology

3.1. Problem Formulation and Notations

Given a reference person image x and a target pose p, our goal is to generate a photorealistic image $\hat{x}$ for that person but in pose p. For arbitrary pose transfer, we simply adopt the commonly-used pose representation to guide such transfer. Specifically, we use 18 human keypoints extracted by a pose estimator [2] as in [23, 34]. The keypoints are encoded into a 18-channel binary heatmap, where each channel is filled with 1 within a radius of 8 pixels around the corresponding keypoint and 0 elsewhere. During training, we consider the image pair (x_1, x_2) (source and target) with their corresponding poses (p_1, p_2). The model takes the triplet (x_1, p_1, p_2) as inputs and tries to generate $\hat{x}_2$ with small error versus target image x_2 in pose p_2.

The proposed dense intrinsic appearance flow consists of two components, namely a flow map $F_{(x_1,x_2)}$ and a visibility map $V_{(x_1,x_2)}$ between image pair (x_1, x_2) to jointly represent their pixel-wise correspondence in 3D space. In the following, we omit the subscript and brief them as F and V for simplicity. Note F and V have the same spatial dimensions as the target image x_2. Assume that u'_i and u_i are the 2D coordinates in images x_1 and x_2 that are projected from the same 3D body point h_i, F and V can be defined as:

$$\begin{aligned} f_i =& F(u_i) = u'_i - u_i, \\ v_i =& V(u_i) = visibility(h_i, x_1), \end{aligned} \tag{1}$$

where $visibility(h_i, x_1)$ is a function that indicates whether h_i is invisible (due to self-occlusion or out of the image plane) in x_1. It outputs 3 discrete values (representing visible, invisible or background) which are color-coded in a visibility map V (see an example in Fig. 3).

3.2. Overall Framework

Figure. 2 illustrates our human pose transfer framework. Given the input image x_1 and its extracted pose p_1, together with the target pose p_2, the flow regression module first predicts from (p_1, p_2) the intrinsic 3D appearance flow F and visibility map V by Eq. (1). Then we use the tuple (x_1, p_2, F, V) for image generation. Note the input image x_1 and target pose p_2 are likely misaligned spatially, therefore if we want to directly concatenate and feed them into a single convolutional network to generate the target image, we can suffer from sub-optimal results. Part of the reason is that the convolutional layers (especially those low-level ones) in one single network may have limited receptive field to capture the large spatial displacements. Some unique network architecture is introduced in [23] to address this.

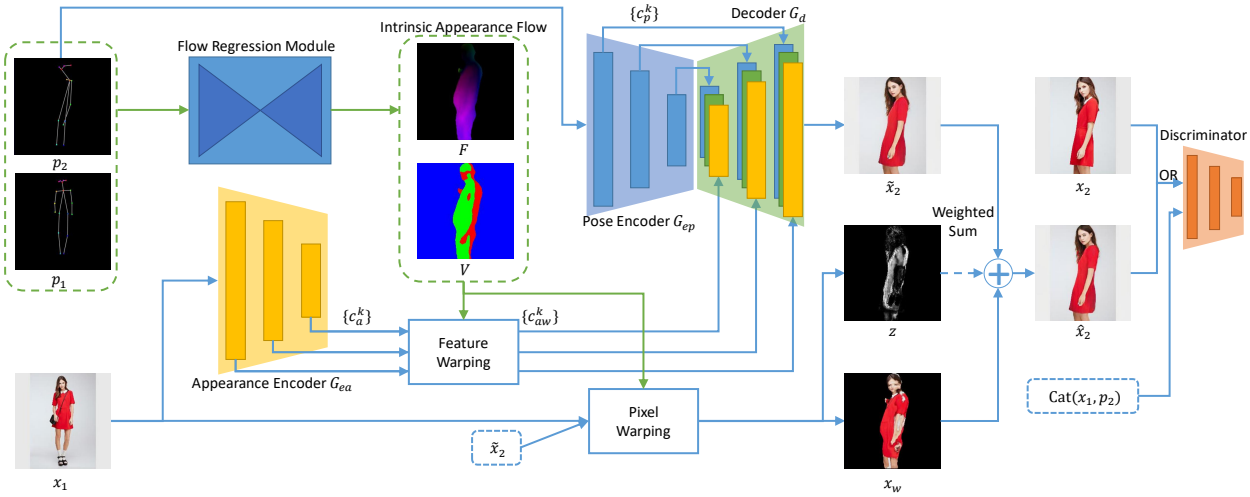

Figure 2: Overview of our human pose transfer framework. With the input image x_1, its extracted pose p_1, and the target pose p_2, the goal is to render a new image in pose p_2. Our flow regression module first generates the intrinsic appearance flow map F and visibility map V, which are used to warp the encoded features $\{c_a^k\}$ from reference image x_1. Such warped features $\{c_{aw}^k\}$ and target pose features $\{c_p^k\}$ can then go through a decoder G_d to produce an image $\widetilde{x}_2$. This result is further refined by a pixel warping module to generate the final result $\hat{x}_2$. Our training objectives include using the PatchGAN [12] to discriminate between (x_1, p_2, x_2) and $(x_1, p_2, \hat{x}_2)$, as well as reconstruction and perceptual losses.

Inspired by [34, 47], we choose to use a *dual-path* U-Net [29] to separately model the image and pose information. Concretely, an appearance encoder G_{ea} and pose encoder G_{ep} are employed to encode image x_1 and target pose p_2 into the feature pyramids $\{c_a^k\}$, $\{c_p^k\}$. Then a feature warping module is proposed to handle the spatial misalignment issue during pose transfer. This module warps the appearance features c_a^k according to our generated flow map F. Meanwhile, some potentially missing pixels in target pose are also implicitly considered by including the visibility map V. Our feature warping function is defined as:

$$c_{aw}^k = W_F(c_a^k, F, V), \tag{2}$$

where W_F is the warping operation detailed in Sec. 3.4, and c_{aw}^k denotes the warped features at feature level k. Then we concatenate warped features $\{c_{aw}^k\}$ and target pose features $\{c_p^k\}$ hierarchically, which are fed to the image decoder G_d through skip connections to generate the target image $\widetilde{x}_2$. Lastly, $\widetilde{x}_2$ is further enhanced by a pixel warping module (Sec. 3.5) to obtain the finial output $\hat{x}_2$.

One of our training objectives is the adversarial loss. We adopt the PatchGAN [12] to score the realism of synthesized image patches. The input patches to the PatchGAN discriminator is either from (x_1, p_2, x_2) or $(x_1, p_2, \hat{x}_2)$. We found the concatenation of (x_1, p_2) provides good conditioning for GAN training.

3.3. Flow Regression Module

Our key module for 3D appearance flow regression is shown in Fig. 3. It is a feedforward CNN that predicts the required appearance flow map F and visibility map V from the pose pair (p_1, p_2). This is similar to the optical flow prediction [4, 10], but differs in that our flow and visibility maps aim to encode 3D dense correspondences not 2D ones in optical flow. For accurate prediction of these two maps, we leverage a 3D human model to synthesize their ground-truth for training.

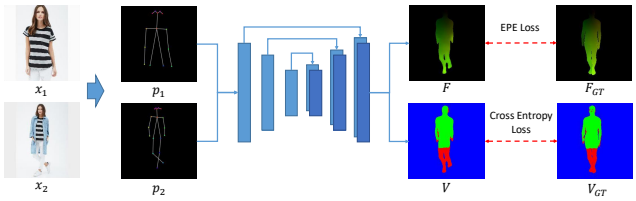

Figure 3: Our appearance flow regression module adopts a U-Net architecture to predict the intrinsic 3D appearance flow map F and visibility map V from the given pose pair (p_1, p_2). This module is jointly trained with an End-Point-Error (EPE) loss on F and a cross-entropy loss on V.

Ground-truth generation. For this purpose, we randomly sample the same-person image pairs (x_1, x_2) from the DeepFashion dataset [20]. We then fit a 3D human model [21] to both images, using the state-of-the-art method [15]. The 3D model represents the human body as a mesh with 6,890 vertices and 13,766 faces. After 3D model fitting, we project them back to the 2D image plane using an image renderer [22]. As indicated by Eq. (1), for the projected 2D coordinate u_j in image x_2, we can identify its exact belonging mesh face in 3D and hence compute the corresponding 2D coordinate u_j' in image x_1 via barycen-

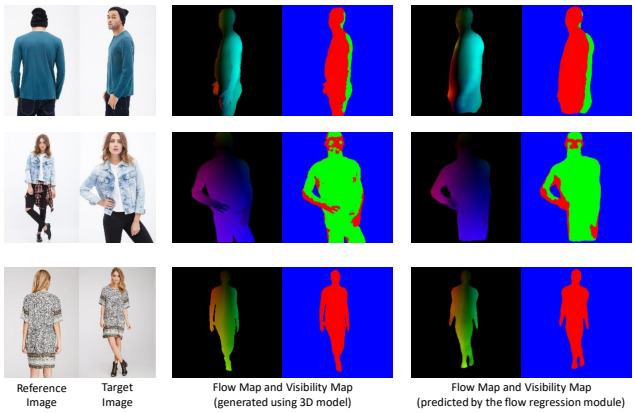

Figure 4: Example 3D appearance flow maps and visibility maps: generated ground-truth (middle) and prediction from our flow regression module (right). The ground-truth rendered from 3D model fitting has occasional errors, *e.g.*, around the overlapping legs in the last row. While our flow regression module can correct the error by predicting from the given pose.

tric transformation. The resulting flow vector is computed as $f_j = u'_j - u_j$. In addition, we can obtain the visibility of each mesh face and thus the entire visibility map V from the image renderer. Fig. 4 (middle) shows some examples of the generated groundtruth flow map and visibility map. One by-product of the 2D image projection is that we can obtain the corresponding 2D pose from the image renderer [15]. We denote such rendered pose as $\widetilde{p}$, and will elaborate its use next.

Network architecture and training. Figure. 3 demonstrates how to train the 3D appearance flow regression module with a U-Net architecture. It takes a pose pair (p_1, p_2) as input and is trained to simultaneously predict the flow map F and visibility map V under the end-point-error (EPE) loss and cross entropy loss, respectively. We noticed that the 3D model fitting process will sometimes cause errors, *e.g.*, when human legs are overlapped with each other, see Fig. 4 (middle, last row). In this case, the synthesized flow and visibility maps $\{F, V\}$ from image-based 3D fitting is not consistent with the groundtruth pose (p_1, p_2) anymore. Hence it is erroneous to train the flow regression from (p_1, p_2) to the un-matched $\{F, V\}$. Fortunately, as mentioned before, we have pose $\{\widetilde{p}_1, \widetilde{p}_2\}$ rendered from the 2D projection process that leads to the corresponding maps $\{F, V\}$. Therefore, we choose to perform regression from the rendered pose $(\widetilde{p}_1, \widetilde{p}_2)$ to $\{F, V\}$, rather than from the potentially un-matched ground-truth pose (p_1, p_2). We found such trained regressor between the rendered pose-flow pair works surprisingly well even when the 3D model is not fitted perfectly. Once our appearance flow regression module finishes training, it is frozen during the training of the overall pose transfer framework. At test time, our flow regression module generalizes well to the given pose p.

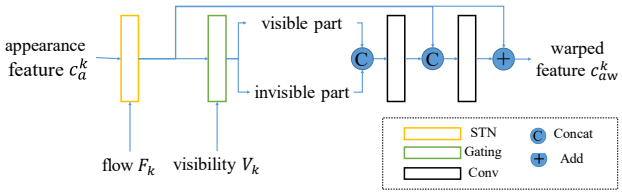

Figure 5: The architecture of feature warping module.

3.4. Flow-Guided Feature Warping

Recall that our 3D appearance flow and visibility maps are generated to align the reference image to the target pose and inpaint the invisible pixels therein. We achieve this by warping the input image features guided by our two maps. The architecture of our feature warping module is illustrated in Fig. 5. The inputs are the image features c_a^k (at feature level k) and the flow and visibility maps (F_k, V_k) resized to match the c_a^k dimensions. We first warp the input features c_a^k by the flow map F_k using a spatial transformer layer [13]. The warped features are then fed into a spatial gating layer, which divides the feature maps into mutually exclusive regions according to the visibility map V_k. Here we do not simply filter out the invisible feature map pixels because they may still contain useful information, like clothing style or body shape. The gated feature maps are passed through two convolutional layers with residual path to get the final warped features c_{aw}^k. Our feature warping module is differentiable allowing for end-to-end training.

3.5. Pixel Warping

As shown in Fig. 2, given the warped features $\{c_{aw}^k\}$, we concatenate them with the target pose features $\{c_p^k\}$ hierarchically. They are both fed to the image decoder G_d through skip connections to render the target image $\widetilde{x}_2$. In our experiments, we found high frequency details are sometimes lost in $\widetilde{x}_2$, indicating the inefficiency of image warping only at feature level. To this end, we propose to further enhance $\widetilde{x}_2$ at pixel level. Similarly, a pixel warping module is adopted to warp the pixels in input image x_1 to the target pose using our 3D appearance flow.

Specifically, we warp x_1 according to the full resolution flow map F to get the warped image x_w. Note x_w contains the required image details from input x_1, but may be distorted because of the coarse flow map and body occlusions. Therefore, we train another U-Net to weigh between the warped output x_w and $\widetilde{x}_2$ at pixel- and feature-level respectively. This weighting network takes x_w, $\widetilde{x}_2$, F and V as inputs and outputs a soft weighting map z with the same resolution of x_w and x_2. The map z is normalized to the range of $(0, 1)$ with sigmoid function. Then the final output

Figure 6: Pixel warping examples. From left to right: the pixel warped image x_w, weighting map z, feature-warped image $\widetilde{x}_2$, final image $\hat{x}_2$ fused with pixel warping, and the ground-truth target image x_2.

x_2^* is computed as a weighted sum of x_w and $\widetilde{x}_2$ as:

$$\hat{x}_2 = z \cdot x_w + (1 - z) \cdot \widetilde{x}_2. \tag{3}$$

Figure. 6 validates the effect of pixel warping. We can see that pixel warping is indeed able to add some high-frequency details that can not be recovered well by our feature warping results. The added details are simply copied from reference image using our intrinsic appearance flow.

3.6. Loss Functions

The goal of our model is to achieve accurate human pose transfer to an arbitrary pose, generating a photorealistic pose-transferred image. This task is challenging due to the large non-rigid deformation during pose transfer and the complex details in human images. Previous works on conditional image generation [12, 38] and human pose transfer [23, 26, 34] utilize multiple loss functions to jointly supervise the training process. In this work we similarly use a combination of three loss functions, namely an adversarial loss $\mathcal{L}_{adv}$, an L1 reconstruction loss $\mathcal{L}_{L1}$, and a perceptual loss $\mathcal{L}_{perceptual}$. They are detailed as follows.

Adversarial loss. We adopt a vanilla GAN loss in the conditional setting in our task, which is defined as:

$$\begin{aligned}\mathcal{L}_{adv}(G, D) =& E_{x_1,x_2}[logD(x_2|x_1, p_2)] \\ &+E_{x_1,x_2}[log(1 - D(G(x_1, p_2)|x_1, p_2))].\end{aligned} \tag{4}$$

L1 loss. Previous work [12] shows L1 loss can stabilize the training process when a target groundtruth is available. Therefore we also enforce an L1 constraint between the generated image and the target image as:

$$\mathcal{L}_{L1}(G) = ||\hat{x}_2 - x_2||_1. \tag{5}$$

Perceptual loss. The work in [14] shows that penalizing L2-distance between feature maps extracted from two images by a pretrained CNN could encourage image structure similarity. We adopt a VGG19 network [36] pretrained on ImageNet [30] as the feature extractor, and use multi-level feature maps ϕ_j to compute perceptual loss as:

$$\mathcal{L}_{perceptual}(G) = \sum_{j=1}^{N} ||\phi_j(\hat{x}_2) - \phi_j(x_2)||_2^2. \tag{6}$$

Our final loss function for image generation is a weighted sum of above terms:

$$\mathcal{L}(G) = \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{L1} + \lambda_3 \mathcal{L}_{perceptual}. \tag{7}$$

4. Experiments

4.1. Dataset and Implementation Details

Dataset. We evaluate our method on DeepFashion dataset (In-shop Clothes Retrieval Benchmark) [20], which contains 52,712 in-shop clothes images and 200,000 cross-pose/scale pairs. The images have a resolution of 256×256 pixels. Following the setting in [34], we select 89,262 pairs for training and 12,000 pairs for testing. We perform additional experiments on Market-1501 dataset [48] and show results in the supplementary material.

Network architecture. Our generator uses a U-Net architecture of $N = 7$ levels. At each feature level, the encoder has two cascaded residual blocks [8] followed by a stride-2 convolution layer for downsampling, while the decoder has a symmetric structure of an upsampling layer followed by two residual blocks. The upsampling layer is implemented as a convolutional layer followed by pixel shuffling operation [32]. There are skip connections between the corresponding residual blocks in the encoder and decoder, and batch normalization [11] is used after each convolutional layer (except the last one). Our discriminator uses the PatchGAN [12] network with a patch size of 70×70 pixels.

Training. We use the Adam optimizer [16] ($\beta_1 = 0.5$,$\beta_2 = 0.999$) in all experiments. We adopt a batch size of 8 and a learning rate of 2e-4 (except for the discriminator which uses learning rate 2e-5). In our experiments we noticed that optimizing the image generator and the pixel warping module separately yields better performance. Therefore we first train the image generator for 10 epochs and freeze it afterwards. Then we add the pixel warping module into the framework and train the full model for another 2 epochs. To stabilize the training, $\mathcal{L}_{GAN}$ is not used in the first 5 epochs.

Table 1: Comparison against previous works on DeepFashion dataset. † indicates the model is unsupervised (no image pairs used in training). ∗ indicates the results are obtained using different data splits, thus cannot be directly compared to ours.

Model	SSIM	IS	FashionIS	AttrRec-k(%)	
				k=5	k=20
UPIS [27]†∗	0.747	2.97	-	-	-
DPT [26]∗	0.796	3.71	-	-	-
DPIG [24]†	0.614	3.228	-	-	-
VUnet [5]†	**0.786**	3.087	-	-	-
PG2 [23]	0.762	3.090	2.639	13.560	30.193
DSC [34]	0.756	**3.439**	3.804	19.017	43.812
Ours	0.778	3.338	**4.898**	**21.065**	**49.044**
Real Image	1.000	3.962	6.518	24.780	61.626

4.2. Evaluation Metrics

Previous works use Structure Similarity (SSIM) [41] and Inception Score (IS) [31] to evaluate the quality of generated images. We report these metrics too in our experiments. However, SSIM is noticed to favor blurry images which are less photorealistic [23]. While IS computed using a classifier trained on ImageNet [30] is not suitable in the scenario where the images have a different distribution than ImageNet images. For these reasons, we introduce two complementary metrics described below.

Fashion inception score. Following the definition in [31], we calculate the inception score using a fashion item classifier, which we refer as FashionIS. Specifically, we finetune an Inception Model [37] on clothing type classification task on [20], which has no domain gap to the images in our human pose transfer experiments. We argue that FashionIS can better evaluate the image quality in our experiments compared to the original IS.

Clothing attribute retaining rate. The human pose transfer model should be able to preserve the appearance details in the reference image, like the clothing attributes like color, texture, fabric and style. To evaluate the model performance from this aspect, we train a clothing attribute recognition model on DeepFashion [20] to recognize clothing attributes from the generated images. Since the groundtruth attribute label of the test image is available, we directly use the top-k recall rate as the metric, denoted as AttrRec-k.

4.3. Quantitative Results

We compare our proposed method against recent works in Table. 1. For SSIM and IS we directly use the results reported in the original papers. We calculate their FashionIS and AttRec-k results using the images generated by the publicly released codes and models. Note that the data splits used in [26, 27] are different from our setting, thus these results are not directly comparable. The results show that our proposed method outperforms others in terms of both FashionIS and AttrRec-k metrics by a significant margin. This proves that our method can generate more realistic images with better preserved details. In terms of SSIM and IS, we also achieve compelling results compared to the state-of-the-art methods.

Figure 7: Qualitative comparison between our method and previous works.

4.4. Qualitative Results

We further visualize some qualitative results in Fig. 7 to show the effectiveness of our proposed method. Because of the introduced 3D intrinsic appearance flow and visibility map, the large spatial displacements and deformations are successfully recovered by our method during pose transfer. We can see that our model generates realistic human image in arbitrary poses and is able to restore detailed appearance attributes like clothing textures.

4.5. User Study

We conduct a user study with 30 users to compare the visual results from our method and the state-of-the-art baseline [34]. The user study consists of two tests. The first one is a "real or fake" test, following the protocol in [23, 34].

Table 2: User study (%) on DeepFashion. R2G indicates the percentage of real images rated as fake, and G2R means the opposite. 'Judged as better' indicates the wining percentage in the comparison test.

Model	R2G	G2R	Judged as better
DSC [34]	9.55	9.24	9.47
Ours	10.01	31.71	90.53

For each method, we show the user 55 real images and 55 fake images in an random order. Each image is shown for 1 second and user will determine whether it is real or fake. The first 10 images are for practice and are ignored when computing results. The second one is a comparison test, in which we show the user 55 image pairs, generated by our method and baseline respectively with the same reference image and target pose, and the user is asked to pick one image with better quality from each pair. The reference image is also shown to make the user aware of the groundtruth appearance. Similar to the first test, the first 5 pairs are for practice. All samples in user study is randomly selected from our test set and shown with full resolution. The results in Table. 2 show that our method generates images with consistently better quality than the baseline, which are confused with real images more often by human judeges.

4.6. Ablation Study

In this section we perform ablation study to further analyze the impact of each component in our model. We first describe the variants obtained by incrementally removing components form the full framework. All variants are trained using the same protocol described in Sec. 4.1.

w/o. dual encoder. This is similar to PG2 [23] that has a U-Net architecture with single encoder and no flow regression module. x_1 and p_2 are concatenated before being fed into the model.

w/o. flow. This model has a dual-path U-Net architecture but without feature warping module. Appearance features $\{c_a^k\}$ and pose features $\{c_p^k\}$ are directly concatenated at corresponding level before sent into the decoder.

w/o. visibility. This model adopts dual-path U-Net generator with a simplified feature warping module, where the gating layer and the first convolution layer in Fig. 5 are replaced with a normal residual block that is unaware of the visibility map V.

Table 3: Ablation study.

Model	SSIM	IS	FashionIS	AttrRec-k(%)	
				k=5	k=20
w/o. dual encoder	0.780	3.173	3.927	19.085	43.377
w/o. flow	**0.783**	3.319	4.119	19.716	44.656
w/o. visibility	0.778	3.260	4.491	20.297	46.591
w/o. pixel warping	0.776	3.281	4.800	20.942	48.391
Full	0.778	**3.338**	**4.898**	**21.065**	**49.044**

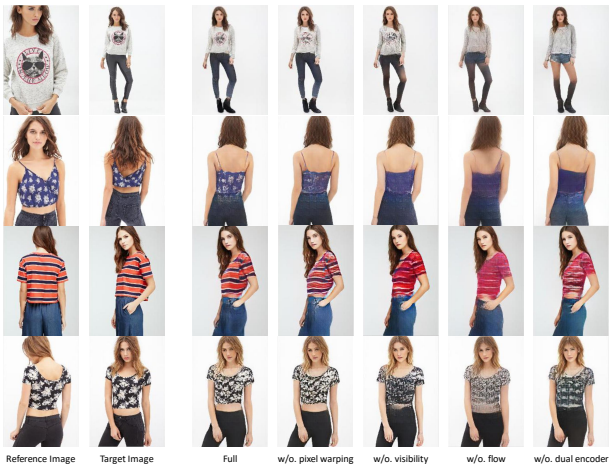

Figure 8: Visualization of ablation study

w/o. pixel warping. This model uses the full generators in Fig. 2 without pixel warping module.

Full. This is the full framework as shown in Fig. 2.

Table. 3 and Fig. 8 show the quantitative and qualitative results of the ablation study. We can observe that all models perform well on generating correct body poses, realistic faces and plausible color style, which yield high SSIM scores. However, our proposed flow guided feature warping significantly improves the capability of preserving detailed appearance attributes like clothing layout and complex textures, which also leads a large increase of FashionIS and AttrRec-k. The pixel warping module further helps to handle some special clothing patterns that are not well reconstructed by the convolutional generator.

5. Conclusion

In this paper we propose a new human pose transfer method with implicit reasoning about 3D geometry of human body. We generate the intrinsic appearance flow map and visibility map leveraging the 3D human model, so as to learn how to move pixels and hallucinate invisible ones in the target pose. A feedforward neural network is trained to rapidly predict both maps, which are used to warp and gate image features respectively for high-fidelity image generation. Both qualitative and quantitative results on the DeepFashion dataset show that our method is able to synthesize human images in arbitrary pose with realistic details and preserved attributes. Our approach also significantly outperforms existing pose- or keypoint-based image generators and other alternatives.

Acknowledgement: This work is supported by SenseTime Group Limited, the General Research Fund sponsored by the Research Grants Council of the Hong Kong SAR (CUHK 14241716, 14224316. 14209217), and Singapore MOE AcRF Tier 1 (M4012082.020).

References

[1] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018. 1

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 3

[3] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. *arXiv preprint arXiv:1808.07371*, 2018. 3

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 4

[5] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018. 1, 3, 7

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2

[7] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 1, 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 6

[9] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 3

[10] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 4

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 4, 6

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 5

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 6

[15] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 4, 5

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2

[18] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. A generative model of people in clothing. In *ICCV*, 2017. 1, 3

[19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[20] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 2, 4, 6, 7

[21] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM TOG*, 34(6):248, 2015. 3, 4

[22] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *ECCV*, 2014. 4

[23] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NIPS*, 2017. 1, 3, 6, 7, 8

[24] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *CVPR*, 2018. 1, 3, 7

[25] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[26] Natalia Neverova, Rıza Alp Güler, and Iasonas Kokkinos. Dense pose transfer. In *ECCV*, 2018. 1, 3, 6, 7

[27] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *CVPR*, 2018. 1, 3, 7

[28] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. Swapnet: Image based garment transfer. In *ECCV*, 2018. 1, 3

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2, 4

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6, 7

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 7

[32] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 6

[33] Raquel Urtasun Dahua Lin Chen Change Loy Shizhan Zhu, Sanja Fidler. Be your own prada: Fashion synthesis with structural coherence. In *ICCV*, 2017. 2

[34] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, 2018. 1, 3, 4, 6, 7, 8

[35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 3

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 7

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 6

[39] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 2

[40] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV*, 2018. 2

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[42] Varun Agrawal Amit Raj Jingwan Lu Chen Fang Fisher Yu James Hays Wenqi Xian, Patsorn Sangkloy. Texturegan: Controlling deep image synthesis with texture patches. *CVPR*, 2018. 2

[43] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes–the importance of multiple scene constraints. In *CVPR*, 2018. 3

[44] Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *CVPR*, 2018. 1, 3

[45] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1710.10916*, 2017. 2

[46] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2

[47] Bo Zhao, Xiao Wu, Zhi-Qi Cheng, Hao Liu, Zequn Jie, and Jiashi Feng. Multi-view image generation from a single-view. In *ACMMM*, 2018. 1, 3, 4

[48] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 2, 6

[49] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 3