Deep Image Spatial Transformation for Person Image Generation

Yurui Ren\textsuperscript{1,2} Xiaoming Yu\textsuperscript{1,2} Junming Chen\textsuperscript{1,2} Thomas H. Li\textsuperscript{3,1} Ge Li\textsuperscript{1,2}

\textsuperscript{1}School of Electronic and Computer Engineering, Peking University \textsuperscript{2}Peng Cheng Laboratory \textsuperscript{3}Advanced Institute of Information Technology, Peking University

\{yrren,xiaomingyu,junming.chen\}@pku.edu.cn tli@ailit.org.cn geli@ece.pku.edu.cn

Abstract

Pose-guided person image generation is to transform a source person image to a target pose. This task requires spatial manipulations of source data. However, Convolutional Neural Networks are limited by the lack of ability to spatially transform the inputs. In this paper, we propose a differentiable global-flow local-attention framework to reassemble the inputs at the feature level. Specifically, our model first calculates the global correlations between sources and targets to predict flow fields. Then, the flowed local patch pairs are extracted from the feature maps to calculate the local attention coefficients. Finally, we warp the source features using a content-aware sampling method with the obtained local attention coefficients. The results of both subjective and objective experiments demonstrate the superiority of our model. Besides, additional results in video animation and view synthesis show that our model spatially transforms the information from sources to targets at the features level. The heat maps indicate the attention coefficients.

1. Introduction

Image spatial transformation can be used to deal with the generation task where the output images are the spatial deformation versions of the input images. Such deformation can be caused by object motions or viewpoint changes. Many conditional image generation tasks can be seen as a type of spatial transformation tasks. For example, pose-guided person image generation \cite{20, 25, 27, 40, 28, 29} transforms a person image from a source pose to a target pose while retaining the appearance details. As shown in Figure 1, this task can be tackled by reasonably reassembling the input data in the spatial domain.

However, Convolutional Neural Networks (CNNs) are inefficient to spatially transform the inputs. CNNs calculate the outputs with a particular form of parameter sharing, which leads to an important property called equivariance to transformation \cite{5}. It means that if the input spatially shifts the output shifts in the same way. This property can benefit tasks such as segmentation \cite{4, 8}, detection \cite{26, 11} and image translation with aligned structures \cite{12, 34} etc. However, it limits the networks by lacking abilities to spatially rearrange the input data. Spatial Transformer Networks (STN) \cite{13} solves this problem by introducing a Spatial Transformer module to standard neural networks. This module regresses global transformation parameters and warps input features with an affine transformation. However, since it assumes a global affine transformation between sources and targets, this method cannot deal with the transformations of non-rigid objects.

Attention mechanism \cite{30, 35} allows networks to take use of non-local information, which gives networks abilities to build long-term correlations. It has been proved to be efficient in many tasks such as natural language processing \cite{30}, image recognition \cite{32, 10}, and image generation \cite{35}. However, for spatial transformation tasks in which target images are the deformation results of source images, each output position has a clear one-to-one relationship with the source positions. Therefore, the attention coefficient matrix between the source and target should be a sparse matrix instead of a dense matrix.

Flow-based operation forces the attention coefficient matrix to be a sparse matrix by sampling a very local source patch for each output position. These methods predict 2-D...
coordinate offsets specifying which positions in the sources could be sampled to generate the targets. However, in order to stabilize the training, most of the flow-based methods \cite{39,3} warp input data at the pixel level, which limits the networks to be unable to generate new contents. Meanwhile, large motions are difficult to be extracted due to the requirement of generating full-resolution flow fields \cite{21}. Warping the inputs at the feature level can solve these problems. However, the networks are easy to be stuck within bad local minima \cite{22,33} due to two reasons. (1) The input features and flow fields are mutually constrained. The input features can not obtain reasonable gradients without correct flow fields. The network also cannot extract similarities to generate correct flow fields without reasonable features. (2) The poor gradient propagation provided by the commonly used Bilinear sampling method further lead to instability in training \cite{14,22}.

In order to deal with these problems, in this paper, we combine flow-based operation with attention mechanisms. We propose a novel global-flow local-attention framework to force each output location to be only related to a local feature patch of sources. The architecture of our model can be found in Figure 2. Specifically, our network can be divided into two parts: Global Flow Field Estimator and Local Neural Texture Renderer. The Global Flow Filed Estimator is responsible for extracting the global correlations and generating flow fields. The Local Neural Texture Renderer is used to sample vivid source textures to targets according to the obtained flow fields. To avoid the poor gradient propagation of the Bilinear sampling, we propose a local attention mechanism as a content-aware sampling method. We compare our model with several state-of-the-art methods. The results of both subjective and objective experiments show the superior performance of our model. We also conduct comprehensive ablation studies to verify our hypothesis. Besides, we apply our model to other tasks requiring spatial transformation manipulation including view synthesis and video animation. The results show the versatility of our module. The main contributions of our paper can be summarized as:

- A global-flow local-attention framework is proposed for the pose-guided person image generation task. Experiments demonstrate the effectiveness of the proposed method.
- The carefully-designed framework and content-aware sampling operation ensure that our model is able to warp and reasonably reassemble the input data at the feature level. This operation not only enables the model to generate new contents, but also reduces the difficulty of the flow field estimation task.
- Additional experiments on view synthesis and video animation show that our model can be flexibly applied to different tasks requiring spatial transformation.

2. Related Work

**Pose-guided Person Image Generation.** An early attempt \cite{20} on the pose-guided person image generation task proposes a two-stage network to first generate a coarse image with target pose and then refine the results in an adversarial way. Essner et al. \cite{2} try to disentangle the appearance and pose of person images. Their model enables both conditional image generation and transformation. However, they use U-Net based skip connections, which may lead to feature misalignments. Siarohin et al. \cite{25} solve this problem by introducing deformable skip connections to spatially transform the textures. It decomposes the overall deformation by a set of local affine transformations (e.g. arms and legs etc.). Although it works well in person image generation, the requirement of the pre-defined transformation components limits its application. Zhu et al. \cite{40} propose a more flexible method by using a progressive attention module to transform the source data. However, useful information may be lost during multiple transfers, which may result in blurry details. Han et al. \cite{7} use a flow-based method to transform the source information. However, they warp the sources at the pixel level, which means that further refinement networks are required to fill the holes of occlusion contents. Liu et al. \cite{18} and Li et al. \cite{16} warp the inputs at the feature level. But both of them need additional 3D human models to calculate the flow fields between sources and targets, which limits the application of these models. Our model does not require any supplementary information and obtains the flow fields in an unsupervised manner.

**Image Spatial Transformation.** Many methods have been proposed to enable the spatial transformation capability of Convolutional Neural Networks. Jaderberg et al. \cite{13} introduce a differentiable Spatial Transformer module that estimates global transformation parameters and warps the features with affine transformation. Several variants have been proposed to improve the performance. Zhang et al. add controlling points for free-form deformation \cite{36}. The model proposed in paper \cite{17} sends the transformation parameters instead of the transformed features to the network to avoid sampling errors. Jiang et al. \cite{14} demonstrate the poor gradient propagation of the commonly used Bilinear sampling. They propose a linearized multi-sampling method for spatial transformation.

Flow-based methods are more flexible than affine transformation methods. They can deal with complex deformations. Appearance flow \cite{39} predicts flow fields and generates the targets by warping the sources. However, it warps image pixels instead of features. This operation limits the model to be unable to generate new contents. Besides, it requires the model to predict flow fields with the same resolution as the result images, which makes it difficult for the model to capture large motions \cite{41,21}. Vid2vid \cite{31} deals with these problems by predicting the ground-truth
flow fields using FlowNet [3] first and then trains their flow estimator in a supervised manner. They also use a generator for occluded content generation. Warping the sources at the feature level can avoid these problems. In order to stabilize the training, some papers propose to obtain the flow-fields by using some assumptions or supplementary information. Paper [24] assumes that keypoints are located on object parts that are locally rigid. They generate dense flow fields from sparse keypoints. Papers [18, 16] use the 3D human models and the visibility maps to calculate the flow fields between sources and targets. Paper [22] proposes a sampling correctness loss to constraint flow fields and achieve good results.

3. Our Approach

For the pose-guided person image generation task, target images are the deformation results of source images, which means that each position of targets is only related to a local region of sources. Therefore, we design a global-flow local-attention framework to reasonably sample and reassemble source features. Our network architecture is shown in Figure 2. It consists of two modules: Global Flow Field Estimator $F$ and Local Neural Texture Renderer $G$. The Global Flow Field Estimator is responsible for estimating the motions between sources and targets. It generates global flow fields $w$ and occlusion masks $m$ for the local attention blocks. With $w$ and $m$, the Local Neural Texture Renderer renders the target images with vivid source features using the local attention blocks. We describe the details of these modules in the following sections. Please note that to simplify the notations, we describe the network with a single local attention block. As shown in Figure 2, our model can be extended to use multiple attention blocks at different scales.

3.1. Global Flow Field Estimator

Let $p_s$ and $p_t$ denote the structure guidance of the source image $x_s$ and the target image $x_t$ respectively. Global Flow Field Estimator $F$ is trained to predict the motions between $x_s$ and $x_t$ in an unsupervised manner. It takes $x_s$, $p_s$, and $p_t$ as inputs and generates flow fields $w$ and occlusion masks $m$.

$$w, m = F(x_s, p_s, p_t)$$

where $w$ contains the coordinate offsets between sources and targets. The occlusion mask $m$ with continuous values between 0 and 1 indicates whether the information of a target position exists in the sources. We design $F$ as a fully convolutional network. $w$ and $m$ share all weights of $F$ other than their output layers.

As the labels of the flow fields $w$ are always unavailable in this task, we use the sampling correctness loss proposed by [22] to constraint $w$. It calculates the similarity between the warped source feature and ground-truth target feature at the VGG feature level. Let $v_s$ and $v_t$ denote the features generated by a specific layer of VGG19. $v_s, w = w(v_s)$ is the warped results of the source feature $v_s$ using $w$. The sampling correctness loss calculates the relative cosine sim-
Figure 3. Overview of our Local Attention. We first extract the feature patch pair from the source and target according to the flow fields. Then the context-aware sampling kernel is calculated by the kernel prediction net. Finally, we sample the source feature and obtain the warped result located at \( l \).

The similarity between \( \mathbf{v}_{s, w} \) and \( \mathbf{v}_t \),

\[
\mathcal{L}_c = \frac{1}{N} \sum_{l \in \Omega} \exp\left(-\frac{\mu(\mathbf{v}_{s, w}^l, \mathbf{v}_t^l)}{\mu_{\text{max}}^l}\right) \tag{2}
\]

where \( \mu(*) \) denotes the cosine similarity. Coordinate set \( \Omega \) contains all \( N \) positions in the feature maps, and \( \mathbf{v}_{s, w}^l \) denotes the feature of \( \mathbf{v}_{s, w} \) located at the coordinate \( l = (x, y) \). The normalization term \( \mu_{\text{max}}^l \) is calculated as

\[
\mu_{\text{max}}^l = \max_{l' \in \Omega} \mu(\mathbf{v}_{s, w}^{l'}, \mathbf{v}_t^{l'}) \tag{3}
\]

It is used to avoid the bias brought by occlusion.

The sampling correctness loss can constrain the flow fields to sample semantically similar regions. However, as the deformations of image neighborhoods are highly correlated, it would benefit if we could extract this relationship. Therefore, we further add a regularization term to our flow fields. This regularization term is used to punish local regions where the transformation is not an affine transformation. Let \( \mathbf{c}_l \) be the 2D coordinate matrix of the target feature map. The corresponding source coordinate matrix can be written as \( \mathbf{c}_s = \mathbf{c}_l + \mathbf{w} \). We use \( \mathcal{N}_n(\mathbf{c}_l, l) \) to denote local \( n \times n \) patch of \( \mathbf{c}_l \) centered at the location \( l \). Our regularization assumes that the transformation between \( \mathcal{N}_n(\mathbf{c}_l, l) \) and \( \mathcal{N}_n(\mathbf{c}_s, l) \) is an affine transformation.

\[
\mathbf{T}_l = \mathbf{A}_l \mathbf{S}_l = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \mathbf{S}_l \tag{4}
\]

where \( \mathbf{T}_l = \begin{bmatrix} x_1 & x_2 & \ldots & x_{n \times n} \\ y_1 & y_2 & \ldots & y_{n \times n} \end{bmatrix} \) with each coordinate \((x_i, y_i) \in \mathcal{N}_n(\mathbf{c}_l, l)\) and \( \mathbf{S}_l = \begin{bmatrix} x_1 & x_2 & \ldots & x_{n \times n} \\ y_1 & y_2 & \ldots & y_{n \times n} \\ 1 & 1 & \ldots & 1 \end{bmatrix} \) with each coordinate \((x_i, y_i) \in \mathcal{N}_n(\mathbf{c}_s, l)\). The estimated affine transformation parameters \( \hat{\mathbf{A}}_l \) can be solved using the least-squares estimation as

\[
\hat{\mathbf{A}}_l = (\mathbf{S}_l^H \mathbf{S}_l)^{-1} \mathbf{S}_l^H \mathbf{T}_l \tag{5}
\]

Our regularization is calculated as the \( \ell_2 \) distance of the error.

\[
\mathcal{L}_r = \sum_{l \in \Omega} \left\| \mathbf{T}_l - \hat{\mathbf{A}}_l \mathbf{S}_l \right\|_2^2 \tag{6}
\]

### 3.2. Local Neural Texture Renderer

With the flow fields \( \mathbf{w} \) and occlusion masks \( \mathbf{m} \), our Local Neural Texture Renderer \( \hat{\mathbf{G}} \) is responsible for generating the results by spatially transforming the information from sources to targets. It takes \( \mathbf{x}_s, \mathbf{p}_t, \mathbf{w} \) and \( \mathbf{m} \) as inputs and generate the result image \( \hat{\mathbf{x}}_t \).

\[
\hat{\mathbf{x}}_t = \hat{\mathbf{G}}(\mathbf{x}_s, \mathbf{p}_t, \mathbf{w}, \mathbf{m}) \tag{7}
\]

Specifically, the information transformation occurs in the local attention module. As shown in Figure 2, this module works as a neural renderer where the target bones are rendered by the neural textures of the sources. Let \( \mathbf{f}_s \) and \( \mathbf{f}_t \) represent the extracted features of target bones \( \mathbf{p}_t \) and
source images $x_s$ respectively. We first extract local patches $\mathcal{N}_n(f_s, l)$ and $\mathcal{N}_n(f_s, l + w^l)$ from $f_s$ and $f_t$, respectively. The patch $\mathcal{N}_n(f_s, l + w^l)$ is extracted using bilinear sampling as the coordinates may not be integers. Then, a kernel prediction network $M$ is used to predict local $n \times n$ kernel $k_l$ as

$$k_l = M(\mathcal{N}_n(f_s, l + w^l), \mathcal{N}_n(f_t, l)) \tag{8}$$

We design $M$ as a fully connected network, where the local patches $\mathcal{N}_n(f_s, l + w^l)$ and $\mathcal{N}_n(f_t, l)$ are directly concatenated as the inputs. The softmax function is used as the non-linear activation function of the output layer of $M$. This operation forces the sum of $k_l$ to 1, which enables the stability of gradient backward. Finally, the flowed feature located at coordinate $l = (x, y)$ is calculated using a content-aware attention over the extracted source feature patch $\mathcal{N}_n(f_s, l + w^l)$.

$$f_{\text{attn}}^l = P(k_l \otimes \mathcal{N}_n(f_s, l + w^l)) \tag{9}$$

where $\otimes$ denotes the element-wise multiplication over the spatial domain and $P$ represents the global average pooling operation. The warped feature map $f_{\text{attn}}^l$ is obtained by repeating the previous steps for each location $l$.

However, not all contents of target images can be found in source images because of occlusion or movements. In order to enable generating new contents, the occlusion mask $m$ with continuous value between 0 and 1 is used to select features between $f_{\text{attn}}^l$ and $f_t$.

$$f_{\text{out}} = (1 - m) \ast f_t + m \ast f_{\text{attn}} \tag{10}$$

We train the network using a joint loss consisting of a reconstruction $\ell_1$ loss, adversarial loss, perceptual loss, and style loss. The reconstruction $\ell_1$ loss is written as

$$L_{\ell_1} = \|x_t - \hat{x}_t\|_1 \tag{11}$$

The generative adversarial framework [6] is employed to mimic the distributions of the ground-truth $x_t$. The adversarial loss is written as

$$L_{\text{adv}} = \text{E}[\log(1 - D(G(x_s, \mathbf{p}_t, w, m)))]$$
\[ + \text{E}[\log D(x_t)] \tag{12} \]

where $D$ is the discriminator of the Local Neural Texture Renderer $G$. We also use the perceptual loss and style loss introduced by [15]. The perceptual loss calculates $\ell_1$ distance between activation maps of a pre-trained network. It can be written as

$$L_{\text{per}} = \sum_i \|\phi_i(x_t) - \phi_i(\hat{x}_t)\|_1 \tag{13}$$

where $\phi_i$ is the activation map of the $i$-th layer of a pre-trained network. The style loss calculates the statistic error between the activation maps as

$$L_{\text{style}} = \sum_j \|G_j^\phi(x_t) - G_j^\phi(\hat{x}_t)\|_1 \tag{14}$$

where $G_j^\phi$ is the Gram matrix constructed from activation maps $\phi_j$. We train our model using the overall loss as

$$L = \lambda_c L_c + \lambda_r L_r + \lambda_{\ell_1} L_{\ell_1} + \lambda_a L_{\text{adv}} + \lambda_p L_{\text{per}} + \lambda_s L_{\text{style}} \tag{15}$$

4. Experiments
4.1. Implementation Details

Datasets. Two datasets are used in our experiments: person re-identification dataset Market-1501 [38] and DeepFashion In-shop Clothes Retrieval Benchmark [19]. Market-1501 contains 32668 low-resolution images $(128 \times 64)$. The images vary in terms of the viewpoints, background, illumination etc. The DeepFashion dataset contains 52712 high-quality model images with clean backgrounds. We split the datasets with the same method as that of [40]. The personal identities of the training and testing sets do not overlap.

Metrics. We use Learned Perceptual Image Patch Similarity (LPIPS) proposed by [37] to calculate the reconstruction error. LPIPS computes the distance between the generated images and reference images at the perceptual domain. It indicates the perceptual difference between the inputs. Meanwhile, Fréchet Inception Distance [9] (FID) is employed to measure the realism of the generated images. It calculates the Wasserstein-2 distance between distributions of the generated images and ground-truth images. Besides,
we perform a Just Noticeable Difference (JND) test to evaluate the subjective quality. Volunteers are asked to choose the more realistic image from the data pair of ground-truth and generated images.

Network Implementation and Training Details. Basically, auto-encoder structures are employed to design our $F$ and $G$. The residual block is used as the basic component of these models. We train our model using $256 \times 256$ images for the Fashion dataset. Two local attention blocks are used for feature maps with resolutions as $32 \times 32$ and $64 \times 64$. The extracted local patch sizes are 3 and 5 respectively. For Market-1501, we use $128 \times 64$ images with a single local attention block at the feature maps with resolution as $32 \times 16$. The extracted patch size is 3. We train our model in stages. The Flow Field Estimator is first trained to generate flow fields. Then we train the whole model in an end-to-end manner. We adopt the ADAM optimizer with the learning rate as $10^{-4}$. The batch size is set to 8 for all experiments.

4.2. Comparisons

We compare our method with several state-of-the-art methods including Def-GAN [25], VU-Net [2], Pose-Attn[40], and Intr-Flow [16]. The typical results of different methods are provided in Figure 4. For the Fashion dataset, VU-Net and Pose-Attn struggle to generate complex textures since these models lack efficient spatial transformation blocks. Def-GAN defines local affine transformation components (e.g., arms and legs etc.). This model can generate correct textures. However, the pre-defined affine transformations are not sufficient to represent complex spatial variance, which limits the

As the subjective metrics may not be sensitive to some artifacts, its results may mismatch with the actual subjective perceptions. Therefore, we implement a just noticeable difference test on Amazon Mechanical Turk (MTurk). This experiment requires volunteers to choose the more realistic image from image pairs of real and generated images. The test is performed over 800 images for each model and dataset. Each image is compared 5 times by different volunteers. The evaluation results are shown in Table 1. It can be seen that our model achieves the best result in the challenging Fashion dataset and competitive results in the Market-1501 dataset.

The typical results of different methods are provided in Figure 4. For the Fashion dataset, VU-Net and Pose-Attn struggle to generate complex textures since these models lack efficient spatial transformation blocks. Def-GAN defines local affine transformation components (e.g., arms and legs etc.). This model can generate correct textures. However, the pre-defined affine transformations are not sufficient to represent complex spatial variance, which limits the
Table 2. The evaluation results of the ablation study.

<table>
<thead>
<tr>
<th></th>
<th>Flow-Based</th>
<th>Content-aware Sampling</th>
<th>FID</th>
<th>LPIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>N</td>
<td>-</td>
<td>16.008</td>
<td>0.2473</td>
</tr>
<tr>
<td>Global-Attn</td>
<td>N</td>
<td>-</td>
<td>18.616</td>
<td>0.2575</td>
</tr>
<tr>
<td>Bi-Sample</td>
<td>Y</td>
<td>N</td>
<td>12.143</td>
<td>0.2406</td>
</tr>
<tr>
<td>Full Model</td>
<td>Y</td>
<td>Y</td>
<td><strong>10.573</strong></td>
<td><strong>0.2341</strong></td>
</tr>
</tbody>
</table>

performance of the model. Flow-based model Intr-Flow is able to generate vivid textures for front pose images. However, it may fail to generate realistic results for side pose images due to the requirement of generating full-resolution flow fields. Meanwhile, this model needs 3D human models to generate the ground-truth flow fields for training. Our model regresses flow fields in an unsupervised manner. It can generate realistic images with not only the correct global pattern but also the vivid details such as the lace of clothes and the shoelace. For the Market-1501 Dataset, our model can generate correct pose with vivid backgrounds. Artifacts can be found in the results of competitors, such as the sharp edges in Pose-Attn and the halo effects in Def-GAN.

The numbers of model parameters are also provided to evaluate the computation complexity in Table 1. Thanks to our efficient attention blocks, our model does not require a large number of convolution layers. Thus, we can achieve high performance with less than half of the parameters of the competitors.

4.3. Ablation Study

In this subsection, we train several ablation models to verify our assumptions and evaluate the contribution of each component.

**Baseline.** Our baseline model is an auto-encoder convolutional network. We do not use any attention blocks in this model. Images $x_s$, $p_t$, $p_s$ are directly concatenated as the model inputs.

**Global Attention Model (Global-Attn).** The Global-Attn model is designed to compare the global-attention block with our local-attention block. We use a similar network architecture as our Local Neural Texture Renderer $G$ for this model. The local attention blocks are replaced by global attention blocks where the attention coefficients are calculated by the similarities between the source features $f_s$ and target features $f_t$.

**Bilinear Sampling Model (Bi-Sample).** The Bi-Sample model is designed to evaluate the contribution of our content-aware sampling method described in Section 3.2. Both the Global Flow Field Estimator $F$ and Local Neural Texture Renderer $G$ are employed in this model. However, we use the Bilinear sampling as the sampling method in model $G$.

**Full Model (Ours).** We use our proposed global-flow local-attention framework in this model.
sparse matrix. However, the Bi-Sample model uses a pre-defined sampling method with a limited sampling receptive field, which may lead to unstable training. Our full model uses a content-aware sampling operation with an adjustable receptive field, which brings further performance gain.

Subjective comparison of these ablation models can be found in Figure 5. It can be seen that the Baseline and Global-Attn model generate correct structures. However, the textures of the source images are not well-maintained. The possible explanation is that these models generate images by first extracting global features and then propagating the information to specific locations. This process leads to the loss of details. The flow-based methods spatially transform the features. They are able to reconstruct vivid details. However, the Bi-Sample model uses the pre-defined Bilinear sampling method. It cannot find the exact sampling locations, which leads to artifacts in the final results.

We further provide the visualization of the attention maps in Figure 6. It can be seen that the Global-Attn model struggles to exclude irrelevant information. Therefore, the extracted features are hard to be used to generate specific textures. The Bi-Sample model assigns a local patch for each generated location. However, incorrect features are often flowed due to the limited sampling receptive field. Our Full model using the content-aware sampling method can flexibly change the sampling weights and avoid the artifacts.

5. Application on Other Tasks

In this section, we demonstrate the versatility of our global-flow local-attention module. Since our model does not require any additional information other than images and structure guidance, it can be flexibly applied to tasks requiring spatial transformation. Two example tasks are shown as follows.

**View Synthesis.** View synthesis requires generating novel views of objects or scenes based on arbitrary input views. Since the appearance of different views is highly correlated, the existing information can be reassembled to generate the targets. The ShapeNet dataset [1] is used for training. We generate novel target views using single view input. The results can be found in Figure 7. We provide the results of appearance flow as a comparison. It can be seen that appearance flow struggles to generate occluded contents as they warp image pixels instead of features. Our model generates reasonable results.

**Image Animation.** Given an input image and a driving video sequence depicting the structure movements, the image animation task requires generating a video containing the specific movements. This task can be solved by spatially moving the appearance of the sources. We train our model with the real videos in the FaceForensics dataset [23], which contains 1000 videos of news briefings from different reporters. The face regions are cropped for this task. We use the edge maps as the structure guidance. For each frame, the input source frame and the previous generated n frames are used as the references. The flow fields are calculated for each reference. The results can be found in Figure 8. It can be seen that our model generates realistic results with vivid movements.

6. Conclusion

In this paper, we solve the person image generation task using deep spatial transformation. We analyze the specific reasons causing unstable training when warping and transforming sources at the feature level. Targeted solution global-flow local-attention framework is proposed to enable our model to reasonably reassemble the source neural textures. Experiments show that our model can generate target images with correct poses while maintaining vivid details. In addition, the ablation study shows that our improvements help the network find reasonable sampling positions. Finally, we show that our model can be easily extended to address other spatial deformation tasks such as view synthesis and video animation.

**Acknowledgements.** This work was supported by Shenzhen Municipal Science and Technology Program (No.JCYJ20170818141146428) and National Natural Science Foundation of China and Guangdong Province Scientific Research on Big Data (No.U1611461).
References


