Guided Flow Field Estimation by Generating Independent Patches

Mohsen Tabejamaat Inria
mohsen.tabejamaat@inria.fr
Farhood Negin
farhood.negin@inria.fr
Francois Bremond
francois.bremond@inria.fr

Abstract

Recent studies have demonstrated the effectiveness of warping in transferring unique textures to the output of the pose transfer networks. However, due to the mutual dependencies of image features and pixel locations, joint estimation of flow map and output image is very likely to get stuck in local minima. Current solution is limited to offline estimation of the maps. However, in this way the flow is generated without interaction with the incarnation parts of the generative model, causing it to struggle with the occlusion parts of samples. To address the issue, we introduce a patch generation module which acts as a mediator between the output values and flow estimations, cutting their mutual dependencies while encouraging the flow maps to merely focus on regions that are not correctly generated by the patch estimations, regions like clothing with unique colors or textures that due to the scarcity of data can not be properly learned during the training phase of the network. Our patch generation module benefits from two individual experts on removing the visible parts of the source sample which disappear in the target view and drawing those invisible parts which appear in the novel view of the sample. Experimental results demonstrate that our method outperforms the state-of-the-art on two well-known databases, Deepfashion and Market1501.

1 Introduction

In this paper, we consider the problem of parser-free pose synthesis [1, 7, 9, 11, 36] which aims to transfer a source sample into a given target pose. As no kind of semantic maps is considered as the input, it is difficult to estimate an accurate warping function that is able to transform each part of the source sample to its corresponding region in the target view. Another problem is with the mutual dependencies of the output sample and the flow map, where it is impossible to accurately estimate the flow map without a correct prediction of the output sample, while the correct estimation of the output sample is already dependent on the correct estimation of the flow field. This causes the network to easily get stuck in local minima.

The single current solution is a two-stage framework [19] that proposes to estimate the maps using an offline pre-training strategy, where an additional network is considered for

extracting the flow map so as to be further used as a prior to the main generative process. This way, the mutual dependency is replaced by a set of pre-defined structures that should be maintained during the process of image generation. However, this comes just at the cost of losing interaction with other parts of the generative model.

To address the issue, we propose the flow maps to adaptively learn from the estimations of the output sample rather than the fix keypoints at the input of the network. To do so, first a holistic estimation of the target pose (along with its texture in invisible parts) is provided by a patch generation module. Then, comparing the estimation with the source sample, a patch transfer module shifts its attention towards the areas that fail to be generated in the previous estimation of the patch generation module. Since the output of the patch generation is completely isolated from the warping maps, even at the point of local minima we still have some gradients which drive the optimization towards a more general solution.

For patch generation, we propose to learn about the source and target samples in a disentangled manner which helps the transfer function be specialized on specific tasks. To do so, target patches are set to be estimated from the same locations in the source sample but through two distinct functions that act as individual experts on the source and target samples. For patch transfer, we utilize an adaptive warping strategy in which the flow map is recursively estimated in interaction with the output of the patch generation module. This way, the estimations of the invisible parts are directly incorporated in the process of flow map estimation which is critically important in realizing a 3D estimation of warping functions despite having operated by 2D functions.

The main contributions of our paper can be summarized as follows: (I) we propose an online strategy for estimating the flow map which benefits from recursive estimation of the output sample rather than a set of sparse keypoints at the input of the network. In contrast to the single current solution [19], it enables the flow map to directly learn from the invisible parts of the samples and also reduces the complexities by paying attention to the regions that can not be properly learned during the training phase of the network. (II) we propose a novel patch generation module which learns to generate the target patches through a set of consecutive operations. This way our module learns to transmute the neighborhoods in their own locations rather than moving them to other parts of the spatial space which has significant difficulties with the limited receptive field of convolutional kernels.

The effectiveness of our method is verified through a series of extensive experiments conducted on two well-known databases, Deepfashion and Market1501 where we outperform the state-of-the-art.

2 Related Work

The topic has recently seen an explosion of scientific works, mostly due to its great potentials in many applications like image animation [21, 22, 23] and virtual reality [12, 17]. In this section, we provide a brief review on the most related work and then clarify the necessity of conducting our proposed method.

Warping: Warping has long been demonstrated to be the most effective way of transferring unique textures in image reconstruction techniques [3, 4, 14, 18, 21, 22, 23, 30]. The idea has widely used in novel view synthesis when the aim is to generate a target view of a static scene [8, 15, 34]. However, simplicity of a static scenario hinders these frameworks to be directly applied for the complex problem of pose generation. For a static scene, it is quite effective to encode a set of simple affine transformations like rotation and translation

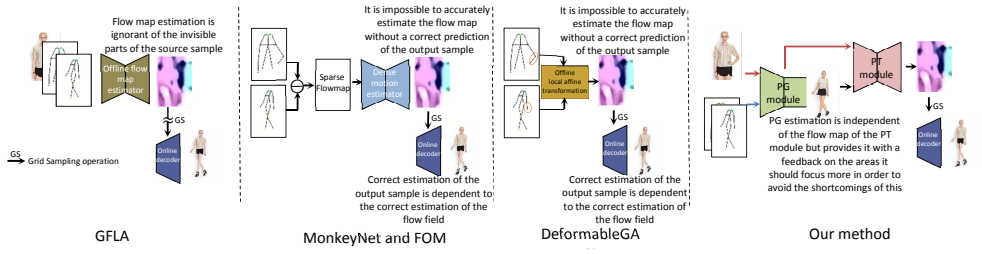

Figure 1: Simplified architectures of flow field estimation in different methods. MonkeyNet and FOM are applicable on video streams

and then apply them for generating a novel view of the sample. However, in the case of a dynamic scene we need to consider some far more complex scenarios in order to deal with the motor complexities of a moving object. For this purpose, the first warping based pose generation method was initialized by aggregating a set of pixel-wise transformations [20]. Later work [22, 23] tried to estimate the flows from differences of some latent codes which are extracted by feeding the keypoints into two individual encoders. These methods strongly suffer from underestimation of manifolds, arising from the sparseness of keypoints.

Progressive Methods: The main idea is to transfer the source patches to their corresponding locations in the target pose [25, 26, 27, 35, 36]. However, despite the claims, the idea has a profound difficulty with moving the patches. Moreover, due to the consecutive concatenation of features, it is difficult for a human observer to interpret the procedure that causes the texture to be washed out during the consecutive updates.

Parser based approaches: Recently, parsers have been widely adopted for image generation tasks [2, 8, 11, 16, 24, 28, 29]. The main idea is built upon the style transfer [8] which proposes to gradually add the source textures to a generative model. For human pose generation, the network is usually provided with the parsing map of the source sample and target map is estimated from its corresponding keypoints [13, 31, 32]. Despite the superior performance of parser based techniques in accurate estimation of clothing shape, their effectiveness for accurate transfer of textures is far inferior than the warping strategies.

In this paper, we propose a combination of two modules named patch generation and patch transfer. Our patch generation module is applied in a hierarchical manner which is similar to the progressive methods, but unlike them it does not seek for moving the patches. Our patch transfer module is based on the warping strategy, but unlike the existing methods, it learns to estimate the flow field based on the adaptive estimations of the output sample to constraint just on the areas that has not been properly generated by the PG module (Figure 1).

3 Our method

Our method consists of three individual modules; Patch Generation (PG), Patch Transfer (PT), and Merging module that are employed along with two additional encoders and one decoder of the model (Figure 2). PG is a fully convolutional module that learns to estimate the whole representation of target samples aimed to further guide the warping maps of the PT module. The second module (PT) is for preserving the locality of textures. The task is

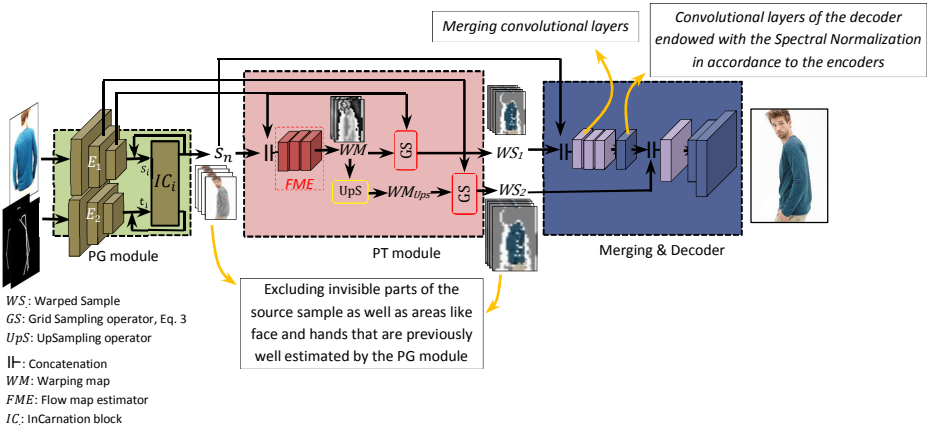

Figure 2: Detailed structure of our proposed method

to keep the clothing textures of the generated image as much similar to the corresponding neighborhoods in the source sample. To avoid any potential overfitting of the decoder, we propose to separate the tasks of blending and decoding of features. To do so, we consider two individual modules, a Merging module and a decoder. Merging is built upon some fully convolutional layers that are placed before the decoder, but the fact that distinguishes between the merging and the decoding part of the network is the different strategies we utilize in their normalization layers, where the encoders and decoder benefit from Spectral Normalization while the Merging is just built upon the Batch Normalization layers.

3.1 Patch Generation Module

3.1.1 Configuration of the module

The module consists of two encoders (E_1 and E_2) and a set of Incarnation blocks (ICs). The encoders learn to project the source image and also the pose representation into a feature space, where the small dimension of embeddings make it easier to combine their characteristics. Pose representation is in fact the volumetric stack of heatmaps concatenated together from the source and target samples. Each heatmap is a Gaussian envelope that is centered on a skeletal keypoint. The intuition behind the generative process of this module can be represented as follows:

Assume that we are already provided with two functions α and β, where α is considered for removing the visible parts of the source sample which disappear in the target view, and β is considered for drawing those invisible parts which appear in the novel view of the sample. This requires both the functions to be calculated from the target pose but making the modifications on the source sample. Obviously, α merely attends to the source pixels. Therefore, it can be directly learned as an attention map that is exclusively applied to the pixels of the source sample, just using the same loss function of the generative process. However, for β, it needs to be informed about the values of the new locations which are merely introduced in the target sample, potentially unavailable when our estimation of the pose is limited to the raw representation of the skeletal points.

To solve the problem, we consider to impose a *constraint* on β in order to endow it with the information about the exclusive locations introduced in the target sample. It is clear that

multiplying the attention map α with the feature map of the source sample and then adding β to the result, we get an estimation of the target sample. Since β is considered to properly distinguish between the newly introduced points in the target sample and the object points of the source image, it should be able to not only distinguish between the points of the target sample from its background but also between different parts of the object in the target view. This implies that multiplying a *function* of β with the target sample will provide us with the pose map of the target sample. But as the *constraint* (mentioned above) we just assume that this *function* can be approximated by a sigmoid function. So, given the raw representation of the target pose t and the source sample s, and the values of α and β, we can estimate the pose of the target sample t_{out} and also the pixel values of the target sample s_{out}.

$$s_{out} = \alpha(t) \otimes f(s) + \beta(t), \qquad t_{out} = (\sigma \circ \beta)(t) \otimes s_{out} \tag{1}$$

where, s and t are respectively referred to as appearance code and pose code of the sample.

It is noteworthy that, the function β should not be confused with an attention map, but it can be expressed in this form, just in order to extract the spatial information of the target sample. Now that we just need to obtain the ideal values of α and β, we can go for optimizing them as part of the general optimization problem of our network.

Since equations in (1) are reciprocally conditioned on each other, in one side we utilize the pose map to estimate the code of the appearance and then benefit from the estimated appearance to update the pose map, it is possible to redefine the equation in the form of a recursive composition that is approximated by a set of local transfer functions. This way, we get more expressive approximations of α and β, defined as a set of simple Incarnation functions $\mathcal{F}^c_{\alpha,i}$ and $\mathcal{F}^c_{\beta,i}$, where i denotes the index of the Incarnation Block and c stands for the channel index of the functions

$$\begin{aligned} s^c_i &= \mathcal{F}^c_{\alpha,i}(t_{i-1}) \otimes f^c_i(s_{i-1}) + \mathcal{F}^c_{\beta,i}(t_{i-1}) \\ t^c_i &= \sigma(\mathcal{F}^c_{\beta,i}) \otimes s^c_i \end{aligned} \tag{2}$$

In fact, each local function $\mathcal{F}_{.,i}$ is a transfer block that receives as the input the estimated pose t_{i-1} and appearance feature map s_{i-1} of the previous block and output the new pose t_i and appearance s_i of the sample. This way, we can generate a compositional transfer function that is expressive enough to represent the complex manifold of images.

3.1.2 What is the need for the PG module?

PG is a hierarchical pose generation module that gradually learns to transfer the values of the source pixels to their corresponding values in the target sample, under a conditioning framework of their poses. Such definition primarily brings to mind the similar idea of the progressive pose transfer [35, 36] and the question that why do we even require a novel algorithm to progressively transfer an image into another while these methods already provide us with the same function, similarly through a set of alternative updates.

The short answer is to avoid overlapping with the task of the PT module. Because our method already benefits from another module whose functionality is merely defined on transferring the patches. Therefore, having another module that also contributes to the task of patch displacement, there would be a mutual dependency that strongly challenges the optimality of reaching towards a global solution of the task. Having this in mind, our PG module was developed so as to just provide a modification on the "value" of pixels without having a role in moving the patches.

Another problem of progressive methods [35, 36] lies in their pose update strategy, whereby a concatenation of the previous pose and appearance is considered to be an updated version of the novel pose. This way, the inference of this concatenation is practically left to the next update of the blocks. Therefore, the next update on appearance will have to manage a collective update on both the appearance and pose characteristics. By repeating this process, it becomes more and more complicated for the module to focus on transferring the textures rather than trying to manage a balanced relation between the pose and textural characteristics of the feature maps. In contrast, our PG is considered to directly extract the pose just by applying an attention map (β which specializes only in incarnating the target poses) on the estimated values of the target sample without any connection to the values of source pixels. This way, the source appearance will be excluded from the process of updating the target pose.

3.2 Patch Transfer

The module is considered to move the patches of the source sample to their corresponding locations in the target pose. The necessity of the module stems from the uniqueness of some clothing patterns that are not frequently present in the training samples, and therefore can not be correctly learned during the training phase of the network.

To create the flow map, we propose to utilize a convolutional module (FME in Figure 2) whose inputs are the feature maps extracted from the encoder E_1 and also the output of the last incarnation block s_n. As the output, the module returns a dual-map whose entries are the locations of pixels along the x and y axes. The map is then utilized in a gride sampling operation to conclude the values of the target features from their corresponding locations in the source feature map. As the gride sampling strategy, we utilize the idea of Spatial Transformer [6] in which the Gride Sampler (GS) projects each point of the source map $s_0(a_i,b_i)$ to the i-th location of the Warped Sample (WS):

$$WS(\mathfrak{i}) = \sum_{(m,n)\in\mathcal{N}(a_i,b_i)} s_0(m,n)max(0,1-|a_i-m|)max(0,1-|b_i-n|) \tag{3}$$

where, $\mathcal{N}(a_i,b_i)$ is a neighborhood of four pixels around the point (a_i,b_i) in the source sample. This way, the sampler is just allowed to copy the pixels at the nearest locations close to (a_i,b_i). By doing so in iterations, it enables to model displacements while also allowing for propagation through the sampling mechanism.

3.3 Merging Module and Decoder

Given two distinct sets of features from the PT and PG modules, we need to determine which parts of these characteristics is more relevant for describing each point of the target sample. The features may be complementary or just one of them is enough to describe the characteristics of a point. To determine how they relates to each other, we propose to encode the neighborhood characteristics where the model finds how to give a priority to the warping features if the neighborhood belongs to a clothing pattern. This can be performed using some convolutional kernels with a receptive field of greater than one, applied on a concatenation of the two sets of features. In addition, as there is a possibility for one set of the feature maps to be descriptive enough in some points, it is necessary for the feature maps to be of the same length. In addition, we utilize the Batch normalization for the convolutional layers

of this module which is different from the Spectrum+Batch normalization that is utilized with the encoders and decoder of our network. This is considered to provide some kind of consistency between the coding parts of our network.

Given the output of the merging module, we utilize a fully convolutional decoder to project the resulting feature maps onto the output space. Our decoder benefits from two skip connections but does not utilize any kind of feature normalizations like AdaIN to avoid any restrictions on the generalization ability of the network, which is a critical issue in tolerating small variations in pose or appearance.

3.4 Training

For training, we consider an adversarial strategy where PG, PT and Merging modules are collectively considered as the generator of the model. In contrast, we utilize a dual-discriminator strategy which is an effective way for incorporating texture-pose consistency into the decision making of the discriminator. This way, the naturalism is only considered in case of those samples whose texture and pose are compatible to the conditional samples of the discriminators. Given this intuition, we define two individual shape and appearance discriminators d_1 and d_2, where d_1 is conditioned on the pose of the target sample and d_2 on the appearance of the source image and consider to train them in a MinMax optimization with the overall generator:

$$L_{adv} = \mathbb{E}\{log[d_1(y,x).d_2(y,p_2)]\} + \mathbb{E}\{log[(1-d_1(G(x,p_1,p_2),x))(1-d_2(G(x,p_1,p_2),p_2))]\} \tag{4}$$

For a better transfer of clothing patterns, we add an additional loss to focus on the comparison of the garment regions. To do so, a perceptual distance is computed over the garment regions of images. Given the binary masks, we calculate the perceptual distance from the VGG embeddings of the masked generated and masked ground truth images. The masks are extracted from the semantic segmentation maps of the images. There is also an alternative to calculate the loss between the mask embeddings rather than the masked images, but in the first way we can incorporate the shape of the garments into the embedding space and consequently into our loss function.

$$L_{pr} = \frac{1}{N}\sum_i \|U_i(y \odot M(y)) - U_i(G(x,p_1,p_2) \odot M(y))\|_1 \tag{5}$$

where $U(.)$ is the i-th feature map of the pretrained VGG19 network and $M(.)$ is the mask image. We also consider the style loss function which measures the correlation between the Gram matrices of the generated and ground truth images.

$$L_{st} = \frac{1}{N}\sum_i \|Q_i(y \odot M(y)) - Q_i(G(x,p_1,p_2) \odot M(y))\|_1 \tag{6}$$

where Q is the Gram matrix we extract from the i-th feature map of the VGG19 network. In addition to the these semantic losses, we also consider the L1 distance between the masked images.

$$L_1 = \|y \odot M(y) - G(x,p_1,p_2) \odot M(y)\|_1 \tag{7}$$

Considering all the functions together, our final loss function is represented as $L_t = \lambda_1 L_{adv} + \lambda_2 L_{pr} + \lambda_1 L_{st} + \lambda_1 L_1$, where $\lambda_.$ is the regularization coefficient of each term.

	reference	parser-based	Deepfashion				Market1501			
			IS↑	SSIM↑	FID↓	LPIPS↓	IS↑	SSIM↑	FID↓	LPIPS↓
ADGAN [13]	CVPR′20	✓	3.3788	0.7716	13.88	0.22	–	–	–	–
PISE[32]	CVPR′21	✓	3.4124	0.7669	9.94	0.20	–	–	–	–
RAN[10]*	CVPR′21	✓	–	0.782	12.24	0.21	–	0.315	23.33	0.27
DeformableGAN[20]*	CVPR′18	–	3.439	0.756	–	–	3.185	0.290	–	–
PATN[33]	CVPR′19	–	3.2019	0.7713	21.73	0.25	3.1604	0.2814	38.36	0.31
SelectionGAN[25]	CVPR′19	–	3.2818	0.7640	32.31	0.27	3.4473	0.3305	104.08	0.34
BiGraphGAN[16]	BMVC′20	–	3.4292	0.7776	24.19	0.24	3.3288	0.3253	36.67	0.30
GFLA[19]	CVPR′20	–	3.4371	0.7673	15.67	0.22	3.1715	0.2803	28.49	0.28
Our method	–	–	3.4621	0.7767	10.80	0.19	3.1919	0.3191	27.69	0.26

Table 1: A comparison between the performance of different pose transfer methods on Deepfashion and Market database, * results reported from the original paper

4 Experiments

In this section, we evaluate the performance of our method in comparison with other state-of-the-art. Evaluations are all conducted on two benchmark databases, Deepfashion [10] and Market1501 [33]. Deepfashion is a high resolution fashion style database, primarily established for online shopping retrieval tasks. To split the samples, 101966 pairs of images are picked up as the training set and 8750 pairs as the testing set of the Deepfashion database. It is noteworthy that the database includes some pairs in which images are not correctly paired together whether in terms of presenting the same clothing or even the same individuals. Market1501 is a low-resolution database, originally collected for monitoring tasks like person re-identification. In this case, we select 263632 pairs of images as the training samples and 12000 pairs as the test samples. The challenge of this database is related to differences in the background and clarity of the paired images which makes it more challenging task to learn a correct transfer function for the main subject.

We benefit from the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. Our learning rate is initialized at 0.0002, which remains constant for $\sqrt{\mathfrak{P}}$ epoches and then declines linearly to zero during another 600 epoches, where $\mathfrak{P}$ is the number of the training pairs. For Deepfashion, images and their corresponding heatmaps are all cropped to the dimensions of 256 x 176, but for Market1501 we crop all the samples to 128 x 64 pixels.

4.1 Quantitative evaluation

In this section, we evaluate our method using four quantitative measures and compare it with the state-of-the-art. The measures are Inception Score (IS), Fréchet Inception Distance (FID), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). SSIM is a fully statistical metric that measures the similarity of the generated samples and their corresponding ground truth images through the statistical features of image patches. This way, the fidelity of the generated image is jointly estimated along with the clarity of pixels. IS, LPIPS, and FID are all considered to measure the semantic similarity of images. However for IS and LPIPS, the metric is a direct measure of the paired embeddings between the generated and ground truth images, while for FID it is a comparison between the distributions of these samples that only measures the realism of the generated images.

Our experimental results are listed in Table 1. As can be seen, there is no absolute winner but our method is quite competitive on all the measures. The point comes from the fact that the overall performance of all these method is a direct measure of the loss functions. For

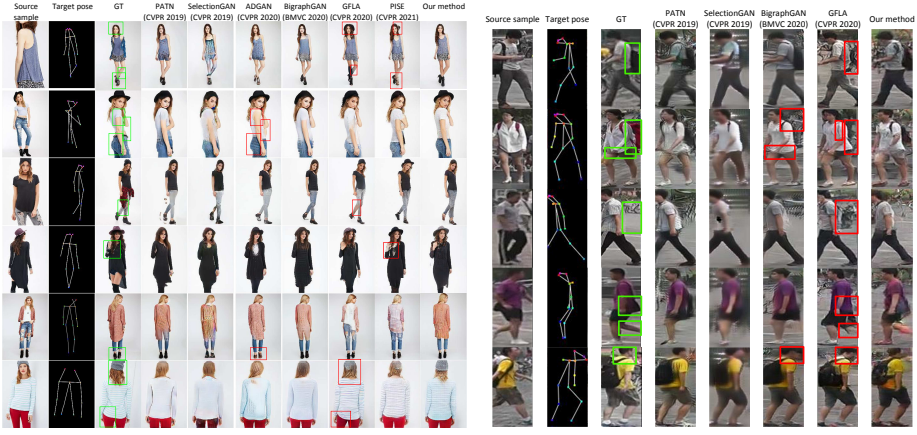

Figure 3: Qualitative comparison between our method and the SOTA, Deepfashion, Market1501

example, the higher the weights of the perceptual loss, the better scores would be achieved for the paired similarity measures like SSIM and IS but this comes at the cost of sacrificing the photo realism of images which is measured by FID score. That is the reason that, it is usually refrained to report all these measures together. BiGraphGAN directly benefits from a intensely weighted perceptual loss, therefore can achieve a fairly good performance on SSIM and IS measures but severely struggles with the visual quality of the generated samples. PISE shows the best result on the FID measure, which can be attributed to the parser maps of this method, but this comes in exchange for the time and effort consuming task of extracting the parsing maps for each test sample. In addition, this method needs to train two individual networks, one for generating the target segmentation map and other for generating the output sample which reduces the scalability of the method. In contrast, our method demonstrates to be quite competitive in all the measures. It achieves the best results on almost all the paired fidelity measures without sacrificing a good visual quality of the generated samples which is further proven in the next section but can be statistically found from the low FID score of the method.

4.2 Qualitative evaluation

For qualitative measure, we visualize a set of samples generated by our method along with their counterparts from the state-of-the-art. All the competing methods have been trained on the same split of the training samples. The results are shown in Figures 3. As can be seen, our method makes a significant improvement in visual quality and semantics of the generated samples, which is beyond the world of numbers presented in the previous section. GFLA has some difficulties with generating correct semantics of images, arising from lack of interaction between an incarnation part and it's local attention module. In general, the fidelity of the textured patterns generated by our method is quite compelling compared to those generated by other methods. The skin color and position of body parts like hands and legs are much more similar to the ground truth images. An interesting results is about preserving the correct garment shape in our method. PISE also has the ability to keep the clothing items of the source samples like wearing a hat, however this method requires a

parsing map which is quite challenging to acquire either in terms of accurate estimation or computational time.

5 Conclusion

In this paper, we proposed a pose generation network which is built upon two individual modules; patch generation and patch transfer. We explored how the generation module can help to learn an online estimation of the warping flow that is critical for the correct transfer of unique textures. We introduced the idea of patch generation that learns to transmute the patches instead of displacing them between different locations of images. This way, we avoid any duplication of tasks that is critical to avoid the mutual dependency of the modules and consequently providing a suboptimal estimation of the warping maps. We argued how employing two individual experts on the characteristics of the source and target samples can help a generative model to estimate the correct patterns of the target patches. We proposed to incorporate the invisible parts of the source sample into the process of flow estimation which makes it to benefit from the information of 3D scenes despite the 2D operation of the warping function. We discussed how online estimation of flow maps restricts the warping features to those critical areas like clothing that due to the scarcity of data can not be properly enveloped during the training phase of the network.

6 Acknowledgments

This work has been supported by the French government, through the 3IA Cote d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

References

[1] Baoyu Chen, Yi Zhang, Hongchen Tan, Baocai Yin, and Xiuping Liu. Pman: Progressive multi-attention network for human pose transfer. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[2] Jaehyeong Cho, Wataru Shimoda, and Keiji Yanai. Mask-based style-controlled image synthesis using a mask style encoder. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5176–5183. IEEE, 2021.

[3] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9026–9035, 2019.

[4] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7543–7552, 2018.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[6] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

[7] Amena Khatun, Simon Denman, Sridha Sridharan, and Clinton Fookes. Pose-driven attention-guided image generation for person re-identification. *arXiv preprint arXiv:2104.13773*, 2021.

[8] Hoang-An Le, Thomas Mensink, Partha Das, and Theo Gevers. Novel view synthesis from single images via point cloud transformation. *arXiv preprint arXiv:2009.08321*, 2020.

[9] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *arXiv preprint arXiv:2106.02019*, 2021.

[10] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

[11] Zhengyao Lv, Xiaoming Li, Xin Li, Fu Li, Tianwei Lin, Dongliang He, and Wangmeng Zuo. Learning semantic person image generation by region-adaptive normalization. *arXiv preprint arXiv:2104.06650*, 2021.

[12] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020.

[13] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5084–5093, 2020.

[14] MR Minar, TT Tuan, H Ahn, P Rosin, and YK Lai. Cp-vton+: Clothing shape and texture preserving image-based virtual try-on. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 2, page 11, 2020.

[15] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 3500–3509, 2017.

[16] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[17] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.

[18] Bin Ren, Hao Tang, Fanyang Meng, Runwei Ding, Ling Shao, Philip HS Torr, and Nicu Sebe. Cloth interactive transformer for virtual try-on. *arXiv preprint arXiv:2104.05519*, 2021.

[19] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7690–7699, 2020.

[20] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018.

[21] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.

[22] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32:7137–7147, 2019.

[23] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. *arXiv preprint arXiv:2104.11280*, 2021.

[24] Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. *arXiv preprint arXiv:2012.04781*, 2020.

[25] Hao Tang, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J Corso, and Yan Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2417–2426, 2019.

[26] Hao Tang, Song Bai, Philip HS Torr, and Nicu Sebe. Bipartite graph reasoning gans for person image generation. *arXiv preprint arXiv:2008.04381*, 2020.

[27] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In *European Conference on Computer Vision*, pages 717–734. Springer, 2020.

[28] Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7870–7879, 2020.

[29] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Dino: A conditional energy-based gan for domain translation. *arXiv preprint arXiv:2102.09281*, 2021.

[30] Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. Dwnet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139*, 2019.

[31] Jichao Zhang, Aliaksandr Siarohin, Hao Tang, Jingjing Chen, Enver Sangineto, Wei Wang, and Nicu Sebe. Controllable person image synthesis with spatially-adaptive warped normalization. *arXiv preprint arXiv:2105.14739*, 2021.

[32] Jinsong Zhang, Kun Li, Yu-Kun Lai, and Jingyu Yang. Pise: Person image synthesis and editing with decoupled gan. *arXiv preprint arXiv:2103.04023*, 2021.

[33] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[34] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016.

[35] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019.

[36] Zhen Zhu, Tengteng Huang, Mengde Xu, Baoguang Shi, Wenqing Cheng, and Xiang Bai. Progressive and aligned pose attention transfer for person image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.