UnWarpME: Unsupervised warping map estimation in real-world scenarios

Mohsen Tabejamaat[a] (mohsen.tabejamaat@inria.fr), Farhood Negin[a] (farhood.negin@inria.fr), François Bremond[a] (francois.bremond@inria.fr)

[a] INRIA, France

Corresponding Author:
Farhood Negin
INRIA, France
Tel: (+33) 602 63 14 10
Email: farhood.negin@inria.fr

UnWarpME: Unsupervised warping map estimation in real-world scenarios

Mohsen Tabejamaat[a], Farhood Negin[a,*], François Bremond[a]

[a] *INRIA, France*

Abstract

This paper introduces a novel approach to warp an RGB image to a new target pose, where it learns to inpaint the invisible parts of the image by sampling pixels from the visible parts. This technique is particularly relevant in applications such as novel pose image generation, where the quality of the generated samples heavily relies on the warped images. Conventional methods typically utilize an affine warping map estimator and learn to estimate the warping map through a downstream image generation task. However, this approach results in an affine function being returned regardless of the complexity of the deformation between the source and target poses. In contrast, our proposed method involves estimating the warping map using a convolutional function, which learns through alternating between two downstream tasks. Initially, the estimation treats the warping as part of an image generation task, similar to existing methods but without the constraint of an affine transformation. This encourages the estimation of complex transformations beyond affine deformations. Subsequently, disregarding the image generation task, we supervise the convolutional warping map to learn any potential affine transformation between the guiding poses of the sample. Our experimental results demonstrate the high effectiveness of our method in preserving image textures while transforming it into highly complex poses.

Keywords: Pose transfer, Image to image translation, Deep spatial transformation

[*] Corresponding author.

Email addresses: `mohsen.tabejamaat@inria.fr` (Mohsen Tabejamaat), `farhood.negin@inria.fr` (Farhood Negin), `francois.bremond@inria.fr` (François Bremond)

1. Introduction

Unsupervised estimation of warping maps[1] is a foundational challenge in computer vision, with diverse applications in 3D reconstruction, image animation, and virtual reality. The main objective is to transform a source image to match a given target pose. This estimation is guided by the location of the sample in the target pose, which can be a set of keypoints or more comprehensive guiding maps such as edge or segmentation maps.

Research in this field has mainly focused on warping based depth estimation from monocular images Poddar et al. (2023); Wang et al. (2024); Nguyen et al. (2024); Yang et al. (2023). This involves utilizing camera parameters to analyze the warping between two RGB images. This information is then employed to estimate a depth map from a single monocular image Minelli et al. (2023); Hou et al. (2021). These methods mainly differ in proposing new attention mechanisms Cao et al. (2022) or considering priors to make depth estimation view-invariant Yoon et al. (2020); Bello & Kim (2024). Some of these methods also incorporate self-supervised learning Johnston & Carneiro (2019) or photometric consistency between frames in video sequences Hasson et al. (2020) to reduce the need for ground truth depth data. Consequently, these advancements have significantly improved the accuracy and reliability of monocular depth estimation models. However, despite advancements, the reverse application of these methods to predict a novel pose of a sample in a different camera location often yields low-quality images, which arises due to the ill-posed nature of this problem, where multiple solutions are possible for each scenario. In contrast, similar to Siarohin et al. (2019b), we consider a scenario where the estimation of the warping map is guided by the keypoint locations of each sample. This causes a reduced ambiguity in potential solutions for each transformation, thereby facilitating near-global optimization of the solution. Existing methods in this context typically assume that the warping function is estimated via supervision from a downstream image generation task while

[1] The term ”unsupervised” indicates that there is no ground truth warping map available to supervise the transformation between the source and target pose of an RGB sample.

constraining the warping function to an affine transformation. While the affine transformation ensures the preservation of texture affinity when transforming images to their target poses, this constraint proves ineffective in handling complex deformations between the source and target pose of the sample. Consequently, existing methods often resort to very large networks to compensate for the limitations of the warping module, necessitating subsequent training of image generation networks with small batch sizes, even in multi-GPU setups.

To tackle this challenge, we propose an unconstrained spatial transformation module. This module acts as a pixel-wise flow estimator, where sampling locations are independently estimated for each pixel of the target image. By doing so, the spatial transformation gains sufficient parameters necessary for accurately estimating complex transformations of the samples. The flow is directly derived from the correlation of the pose maps, enabling precise encoding of both short and long-range displacements that cannot be adequately captured by a simple convolutional network.

Our method not only learns to displace the visible patches to their new position but also learns to introduce the invisible parts to the warped images. To achieve this, we alternate between two distinct tasks. Firstly, we supervise the convolutional flow estimator to estimate an affine function for any linear transformation between the guiding maps of the samples. This supervision is independent of any downstream task, ensuring the effective contribution of the guiding maps in estimating the warping map regardless of the content of input images. Secondly, we supervise the flow estimator to create a warped image that best fits the RGB sample in the target pose. Benefiting from an unconstrained convolutional warping map estimator, with this second supervision, it can learn any complex deformation between the pose of the samples. One main advantage is that the warping can introduce novel parts in the warped image that are invisible in the source pose of the sample, an advantage not possible with existing methods that constrain the warping function itself to be an affine transformation.

In summary, the main contributions of our paper are: (1) we introduce a pixel-wise flow estimator that independently estimates sampling locations for each pixel in the target image, enabling accurate estimation of complex transformations between the source pose and the target pose of samples. (2) Unlike methods constrained to affine transformations, our method learns to introduce novel parts in warped images that are not visible in the source pose, thereby enhancing the fidelity and completeness of the transformation process.

Additionally, we demonstrate the effectiveness of our method using two well-known databases: Deepfashion and Market-1501. Our approach outperforms existing state-of-the-art methods for unsupervised warping estimation using guiding poses.

The remainder of the paper is structured as follows: Section 2 provides a concise overview of related work in the field. Section 3 presents the preliminaries relevant to our approach. Section 4 describes our proposed methodology. Section 5 discusses implementation details and presents the results of our research, while Section 6 concludes the paper.

2. Related work

In this section, we review existing methods relevant to our proposed approach, categorized into three main groups that utilize warping maps estimation for various tasks: Pose Transfer, Flow Estimation, and Geometric Matching.

Pose transfer: There are two different approaches for generating a novel pose of a human in an RGB image: (i) parametric methods Thies et al. (2016); Corona et al. (2021) that learn to fit a 3D model on the RGB sample and then render the model from a novel view point with a different pose. They have been already applied to face Thies et al. (2016) and body Corona et al. (2021). Despite the memory advantage and remarkable control over the pose of the sample, these methods suffer from certain disadvantages including the difficulty of fitting a 3D model on a 2D image. Moreover, they have great difficulty with the lack of fine-grained details in the generated samples. This arises from the generic 3D model that is shared between all the RGB samples. (ii) non-parametric models Liu et al. (2021); Tang et al. (2020); Zhang et al. (2021); Lv et al. (2021); Tang & Sebe (2021); Tabejamaat et al. (2021, 2024). These methods are also known as exemplar-based techniques. The main idea is to remove the need for a predefined 3D model. Rather, they propose to reconstruct the new images using a deep neural network that is guided by the target pose of the samples. The pose can be represented by a set of keypoints Ren et al. (2020); Tang et al. (2020); Zhang et al. (2021), edge maps Ren et al. (2020), or segmentation maps Liu et al. (2021); Zhang et al. (2021). Unlike the parametric models, these methods are proven to be more effective in generating the fine-grained textures of the samples. Despite promising results, they have some difficulties to generate the textures that are not among the training samples. This hinders the scalability of these methods for the vast majority of human garments. The challenge can be well addressed by introducing a warping module in the latent space of these networks. This module is supervised by an estimation of the flow map between the pose of the source and the target samples Ren et al. (2020); Siarohin et al. (2019b,a, 2018); Zhang et al. (2021). However, these methods have significant difficulties with an accurate estimation of the flow maps, especially between two completely different poses.

Flow estimation: Unsupervised flow estimation Ren et al. (2017); Sabour et al. (2021); Stone et al. (2021); Luo et al. (2021) is one of the most difficult tasks in computer vision, especially when estimated from two very sparse sets of keypoints. The task is usually simplified by reducing the flow function to a simple parametric model like first-order affine transformation Siarohin et al. (2018, 2019b,a) or thin spline transformation Zhao & Zhang (2022). In this case, the new position of each pixel is determined by applying a simple algebraic operation on its current coordinates. There are some other strategies to estimate the flow field from the UV maps AlBahar et al. (2021); Sarkar et al. (2021), or 3D meshes Li et al. (2019). Compared to the keypoints, they provide a fine description of the custom shape, which facilitates the estimation of the flow maps. However, these representations are not easily accessible for an unseen pose that does not exist among the training samples. By contrast, keypoints can be easily provided by drawing a few dots on an empty scene which is highly convenient for many applications, making it the *de facto* standard of the pose transfer networks.

There are also some strategies that estimate a flow map based on the supervised strategies Rocco et al. (2017); Truong et al. (2020) in which the transformation between the source and the target samples are already given to the network.

Geometric matching: These techniques focus on estimating corresponding locations between two RGB imageswith significant disparities in their viewpoints. Strategies, such as CACIM and GoCor, introduce attention mechanisms across multiple stages of geometric matching networks to extend their matching process to a multi-resolution framework. This facilitates comprehensive comparison, considering all feasible shifts between the images. Additionally, these methods often leverage a hinge loss function to enforce proximity of true descriptors while distancing false ones.

There exist some other strategies that benefit from prior knowledge about the geometry of images. These approaches usually employ information regarding camera angles to learn a 3D reconstruction of the images, and subsequently render scenes to match the second pose. Stereo algorithms are another direction of research which learn to reconstruct a 3D scene between two sterio images with limited difference in their viewpoints.

In contrast to existing methods, we introduce a nonparametric approach for unsupervised flow estimation capable of handling any complex deformations between the pose of the samples. To the best of our knowledge, our model is the first to leverage warping flow to generate invisible parts of samples. This enables the estimation of visible patches while simultaneously generating invisible ones within a single warping module.

3. Unsupervised warping

Spatially transforming a source image to a new pose necessitates an accurate warping map that precisely locates each pixel in the target pose. This task is typically performed through a warping module integrated into a pose transfer network. The fidelity of texture transfer from the source image to its corresponding locations in the target pose directly influences the number of parameters needed in subsequent stages of the pose transfer network. In this overview, we provide an outline of existing warping modules commonly employed within pose transfer networks.

3.1. First order motion model

FOMM first estimates a set of discrete affine transformations. Each transformation is assumed to represent the motion of each body part, $\mathfrak{A}^{\mathfrak{v}} \in \mathbb{R}^{2\times3}$, $\mathfrak{v} = 1, ..., \mathfrak{t}$, where $\mathfrak{t}$ stands for the number of transformation. This transforms the entire pixels to a reference frame and then from the reference frame to the target pose, $\mathfrak{A}^{\mathfrak{v}} : \mathfrak{A}^{\mathfrak{v}}_{s,r} \longrightarrow \mathfrak{A}^{\mathfrak{v}}_{r,t}$, where $\mathfrak{A}^{\mathfrak{v}}_{s,r}$ is the transformation from the source frame to the reference frame and $\mathfrak{A}^{\mathfrak{v}}_{r,t}$ is the transformation from the reference frame to the target frame. $\mathfrak{A}^{\mathfrak{v}}$ is computed from a set of keypoints, respectively estimated for a pair of source and target images. The keypoints are estimated using an autoencoder which outputs $\mathfrak{K}$ heatmaps $\mathfrak{M}_1, ..., \mathfrak{M}_{\mathfrak{K}}$. Each heatmap is headed by a softmax to give more importance to just one pixel of the scene, $\mathfrak{M}_{.}(\mathfrak{i}, \mathfrak{j}) \in [0, 1]$, where $\mathfrak{i} \in [0, \mathfrak{H} - 1]$, $\mathfrak{j} \in [0, \mathfrak{W} - 1]$. Softargmax is then used to compute the location of the maximum response in each of the heatmaps. These points are directly used to determine the translation component of the affine transformations.

The remaining components of the transformations are directly estimated using another autoencoder. A single warping map is finally estimated using weighted pooling of the affine transformations.

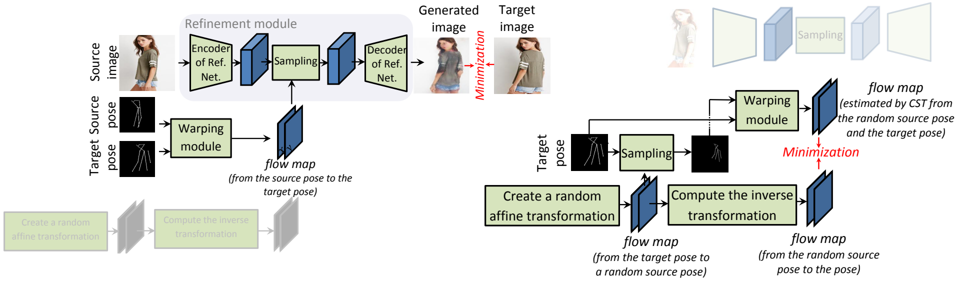
Refinement module
Source image
Encoder of Ref. Net.
Sampling
Decoder of Ref. Net.
Generated image
Target image
Minimization
Source pose
Target pose
Warping module
flow map
(from the source pose to the target pose)
Create a random affine transformation
Compute the inverse transformation
Sampling
Warping module
flow map
(estimated by CST from the random source pose and the target pose)
Target pose
Sampling
Minimization
Create a random affine transformation
Compute the inverse transformation
flow map
(from the target pose to a random source pose)
flow map
(from the random source pose to the pose)

Figure 1: Overall pipeline of our method, trained by alternating between two minimization tasks, depicted on the left and right sides of the figure.

3.2. PCA-based motion estimation

PCAME is the warping module of MRAA. The process is highly similar to FOMM, where the translation components of the affine transformations are computed as the same, but the rotational and scaling components are computed by the singular value decomposition of each heatmap. This provides an explicit representation of the geometry for each part of the moving object in the scene.

However, this method does not fully represent an affine transformation as it does not consider a shear component, reducing the ability of the method to capture any transformation that includes a shear transformation.

3.3. Thin-plate spline motion model

Unlike FOMM and PCAME, Thin Plate Spline (TPS) is a nonlinear strategy for estimating the warping map between two sets of keypoints. This allows for estimating more complex transformations. Similar to FOMM and PCAME, TPSMM benefits from K discrete transformations. The υth transformation is formulated as follows:

$$\mathfrak{T}_{\mathfrak{v}}(x, y) = \mathfrak{A}^{\mathfrak{v}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \sum_{\mathfrak{i}=1}^{\mathfrak{N}} \mathfrak{w}_{\mathfrak{i}}^{\mathfrak{v}} \mathfrak{r} \log(\mathfrak{r}) \tag{1}$$

where $r = \| \begin{bmatrix} x_{\mathfrak{i}}^{p} \\ y_{\mathfrak{i}}^{p} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \|^2$, $\mathfrak{A}$ and $\mathfrak{w}^{\mathfrak{v}}$ are the TPS coefficients. To determine the coefficient, the warping map is bounded by physical bending energy, so that the transformation will be subjected to only slight deformations. (x, y) and $(x_{\mathfrak{i}}^{p}, y_{\mathfrak{i}}^{p})$ respectively represent the coordinate of each pixel and the coordinate of a keypoint. Compared to FOCMM and PCAME, TPS is more sensitive to the number of landmarks. Similar to PCAME, TPS considers an extra affine transformation for modeling the background movement.

3.4. sampling correction

This method estimates the warping function using a convolutional network, constrained by a sampling correctness loss. Given two sets of keypoints represented by heatmaps, the network calculates the similarity between the first heatmap and the deformed second heatmap at an intermediate layer of the VGG network. The sampling correctness quantifies the overall similarity as the sum of the cosine similarities between corresponding patches of the target and the warped source heatmaps. Each local similarity is normalized by the maximum similarity among local patches. Additionally, this flow field estimation is encouraged to approximate an affine transformation through another loss function.

4. Our method

Our main objective is to spatially transform an RGB image to a specified target pose, defined by the locations of body keypoints. To achieve this, we introduce a fully convolutional strategy in which the warping map is estimated using a convolutional network. Unlike affine transformation-based strategies, our approach enables the warping module to estimate complex deformations. Our model receives a combination of the source image, the source pose map, and the target pose map, and learns to estimate the (x, y) coordinates of each pixel in the target pose. Each pose map is an all-zero surface where the location of each body joint is indicated by a small Gaussian envelope.

The conventional use of warping modules is to displace image pixels to their target positions. However, our objective is distinct from these traditional methods. We aim to develop a warping method that not only displaces pixels but also utilizes warping to inpaint parts that are not visible in the source image. Our method learns to inpaint these parts in a completely unsupervised manner, meaning it does not require any additional guiding map to indicate the location of these invisible parts. Consequently, we avoid any confusion regarding the origin of a decrease in the loss function. It clarifies whether the decrease in the loss function happens because we successfully move a visible part of the image to its correct position in the target pose or because it successfully fills in the parts of the image that are missing in its source pose. This enhances the clarity of the learning process and facilitates more effective training of the network compared to direct supervision of pixels.

Figure 1 illustrates the overall pipeline of our method, which consists of two distinct modules: (1) a warping module $\mathcal{W}$ responsible for learning to estimate a flow map between the pose of the source sample and the target pose, and (2) a refinement network $\mathcal{R}$ designed to receive the warped version of the source image and fill in the missing part of this image.

4.1. Warping module

4.1.1. Flow estimation

Given a source pose $p_s \in \mathbb{R}^{h\times w}$ and a target pose $p_t \in \mathbb{R}^{h\times w}$, our warping module, $\mathcal{W}$, estimates a flow map $f_{s,t} \in \mathbb{R}^{2\times h\times w} = \mathcal{W}(p_s, p_t)$. This flow map is later used to spatially transfer the source image $I_s \in \mathbb{R}^{3\times h\times w}$ to its target pose. Our estimation involves 7 different steps, ultimately producing a flow map capable of generating the unseen parts of the source sample I_s while displacing its visible parts to their new locations in the target pose.

1. Projecting p_s and p_t to a feature space using a single convolutional network: $p_s \rightarrow z_s, p_t \rightarrow z_t$[2]
2. Creating a correlation tensor from z_s and z_t using the pairwise similarity of their pixels: $(z_s, z_t) \rightarrow C$
3. Flagging a single pixel of z_s as the corresponding point for each pixel of z_t. To do so, we multiply Q by an attention tensor γ: $Q_\gamma \rightarrow \gamma C$
4. Estimating the flow map by applying a set of convolutional layers on Q_γ: $Q_\gamma \rightarrow f_{s,t}$
5. Applying $f_{s,t}$ to warp the source image. Then, we use a refinement network to estimate the target sample from this warped image. The estimation is encouraged by minimizing the distance between the output of the refinement network and the target sample.
6. Enforcing steps 1 to 4 to estimate an affine function if p_t is a linear transformation of p_s.
7. Alternating between the last two steps. In this way, the network leans to keep the vicinity of pixels in those parts whose movements from p_s to p_t are linear (i.e., can be approximated by an affine transformation) and use a non-constrained estimation for the remaining parts.

Creating a guided correlation map (corresponding to steps 1 to 3):

The flow is directly estimated from the correlation between the source pose p_s and the

[2] z_s and z_t are three dimensional feature maps

target pose p_t. Through correlation, our goal is to establish a one-to-one correspondence between each target pixel and its corresponding point in the source sample. This approach provides two advantages: (1) every point in the target pose is compared with all pixels in the source pose, enabling the network to handle large displacements of samples; (2) the network assigns each pixel to its relevant area in the source sample, even for pixels corresponding to invisible parts of the source image. In such cases, the sampling point is selected from an area with the most similar texture.

To do so, both the source and the target poses are first projected to a feature space: $p_s \xrightarrow{\mathcal{M}} z_s \in \mathbb{R}^{m \times h_1 \times w_1}$, $p_t \xrightarrow{\mathcal{M}} z_t \in \mathbb{R}^{m \times h_1 \times w_1}$, where $\mathcal{M}$ is a fully convolutional network, h_1 and w_1 are the spatial dimension of the feature maps, and m is the dimensionality of the features. Thus, we have two dense feature maps which are later used for creating a *dense* correspondence between the source and the target poses of the sample. The correlation is computed based on the pixel-wise scalar product of the feature maps which is represented as follows:

$$C(i,j,,v,u) = \sum_{\rho=1}^{m} z_s(i,j)[\rho] z_t(v,u)[\rho] \tag{2}$$

where $z_s(i,j) \in \mathbb{R}^m$ and $z_t(v,u) \in \mathbb{R}^m$. $z_s(i,j)[\rho]$ indicates the ρ-th element of the feature vector in the spatial location (i,j) of the feature map z_s.

Next, each pixel in the target pose is assigned to exactly one pixel in the source sample. To achieve this, C is filtered to retain maximum similarity for each target pixel (v,u), while suppressing values in other parts. This filtering employs an attention mechanism that emphasizes locations with maximum similarities, defined as $Q_\gamma = \gamma C$. However, since the dot product combines angle and magnitude, its unbounded results can lead to sensitivity regarding sample magnitude. To address this, we normalize the features at each spatial location (u,v) of Q_γ:

$$Q_\gamma(i,j,,v,u) = \frac{Q_\gamma(i,j,v,u)}{\sqrt{\sum_{i,j=1}^{h_1,w_1} \left(Q_\gamma(i,j,v,u)\right)^2}} \tag{3}$$

Estimating a flow map from a correlation tensor (corresponding to step 4):

The flow is estimated by applying a series of convolutional layers to the refined correlation tensor Q_γ. First, Q_γ is rasterized so that each position (v,u) is vectorized as (i,j).

The resulting tensor is then processed through convolutional layers: $f_{s,t} = \mathcal{C}_l(Q_\gamma)$, where $\mathcal{C}_l$ denotes convolutional filtering. This process estimates the flow at each target position by correlating it with all positions of the source pose, ensuring effective capture of long-range displacements.

Estimating pixel visibility in an unsupervised manner (corresponding to steps 5 to 7):

Up to this point, we have a non-parametric flow estimator, where the sampling locations are individually estimated for each pixel of the target sample. However, such estimation has no constraint to keep the vicinity of the pixels[3], which is important when reconstructing the textures that are visible in both the source and the target poses of the sample. In contrast, binding the estimation to keep the vicinity for the entire 2D space restricts the ability of the flow map to introduce any novel region that is not present in the source sample, especially for the regions with a novel shape, as the novel shapes can not be reconstructed by duplicating the visible parts of the sample.

To avoid this problem, we enforce the flow estimator to keep the vicinity of the pixels just in the visible areas. We approximate these regions with the areas that are displaced using a linear (affine) transformation. Then, for the remaining areas, the flow is estimated without applying any constraint on the pixel-wise estimator. This way, the vicinity of pixels, preserved by the affine transformation, is just applied to the areas that are visible in both the source and the target poses of the sample.

We implement this strategy in an unsupervised manner. To do so, we make steps 1 to 4 to alternate between two different tasks: (1) learning an affine transformation if the target pose is a linear transformation of the source pose, (2) estimating a warping map that best reconstructs the target image from the warped version of the source image.

For the first task, we first generate a random affine transformation using a few random parameters:

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} (-1)^{a_1} + a_2 & a_3 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{4}$$

[3] Here, we consider the vicinity in a large neighborhood which is more related to preserving the overall integrity of patches in the source sample

where $a_1, ..., a_5$ and b_1, b_2 are all random variables. If we sample the pixels of the target pose p_t using the generated affine transformation, it provides us with a novel pose p_{st} whose displacement is linear with respect to the target pose. This pose is considered as the source pose of task (1). In practice, we restrict the random values $a_1, ..., b_2$ so that the resulting transformation does not make a big difference in the verticality of the human skeletons.

Then, we generate the inverse flow of this transformation using the following equation:

$$\begin{bmatrix} v \\ u \end{bmatrix} = \Big(\begin{bmatrix} (-1)^{a_1} + a_2 & a_3 \\ a_4 & a_5 \end{bmatrix} \Big)^{-1} \Big(\begin{bmatrix} i \\ j \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \Big) \tag{5}$$

This transformation is the flow map that transfers the source sample to the target pose. We require the target pose to be real. This avoids learning irrelevant mappings to a huge number of unrealistic poses that never happen in the real world. This is the reason for calculating the inverse flow and applying it on a randomly generated source pose. We refer to the flow map made by this affine function as $s_f = \{(v, u) | v \in \{1, ..., h_1\}, u \in \{1, .., w_1\}\}$.

Then, we minimize the distance between $f_{s,t}$ and s_f when the warping module receives p_{st} and p_t as its input samples. In this way, we enforce the pixel-wise flow map $f_{s,t}$ to be a linear function and therefore to keep the integrity of the neighboring pixels. Next, we need to avoid this integrity to be applied to the areas whose displacements are not linear (the pixels that are invisible in the source image). To do so, we simply feed the module with the original p_s and p_t, estimate the pixel-wise flow map $f_{s,t}$, and then utilize it to warp the source sample. This warped image is then used to reconstruct the target sample. Since we do not use any *guiding map* (in concatenation with our warped image), this reconstruction requires the network to include all the invisible areas in the warped image. This is feasible due to the pixel-wise estimation of the flow map that does not apply the same function to the entire space. Therefore, a point can be sampled from any part of the source image regardless of the sampling location for its neighboring pixels. In this way, the pixels can take any arbitrary shapes, which is necessary for a complete match between the warped image and the target sample.

Finally, we only need to enforce the flow estimation to differentiate between the visible and the invisible parts of the sample and to restrict the affinity of the transformation to the visible areas. This is performed by alternating between task (1) and task (2). For task (1), we already enforced $f_{s,t}$ to be an affine function when the entire p_t is a linear transformation of the entire p_s. However, by nature $f_{s,t}$ is a pixel-wise transformation. When we alternate between two tasks, one of which is a pixel-wise transformation and the other one is the same function for the whole pixels, the network learns to apply the holistic one in a local manner. Therefore, it learns to apply the affine transformation on the local neighborhoods whose displacements[4] are linear (*visible* pixels). It means, the network learns to keep the vicinity of pixels in the local neighborhoods that are *visible* even if they are surrounded by the *invisible* areas.

4.2. Refinement network

Even if the warping module can ideally generate the invisible parts of a sample, it has no prior information regarding the background of the images. Additionally, warping has significant difficulties in generating realistic faces or hands. Therefore, the refinement module $\mathcal{R}$ is used to inpaint these regions and generate a photorealistic sample from the warped image. To do this, we utilize a few convolutional layers without any skip connections. The module receives the source image along with the warping map generated by the flow estimation module. It first relies on a convolutional encoder to project the input image into a low-dimensional feature space, $\theta_s = E_{ref}(I_s)$, then utilizes the flow map to warp this resulting feature map, $\eta_s = \mathcal{T}(\theta_s, f_{s,t})$, where $\mathcal{T}$ is the sampling operation. Next, we use a convolutional decoder to reproject the warped feature map η_s to the original dimension, $y_o = D_{ref}(\eta_s)$. y_o is the image that estimates the target sample. Finally, $f_{s,t}$ is used for direct sampling from the pixels of the source image, directly transferring the image to its target pose.

The encoder-decoder configuration allows adding more learnable parameters to the refinement process, which increases the overall performance of the network in generat-

[4] here, we mean the transformation between the source and the target pose of the sample

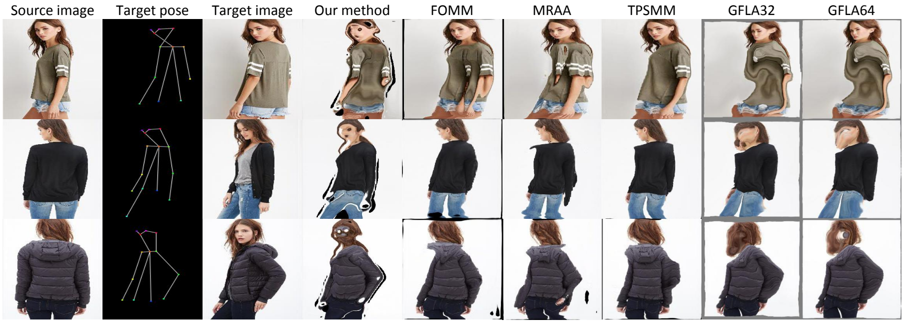

Figure 2: Visual comparison of our method and the state-of-the-art techniques (**these images are generated by direct sampling of pixels from a source image and not using a convolutional network or a generative model. For additional results, please refer to Figures 7-10 at the end of the paper.**).

ing fine-grained textures. In practice, the module is responsible for generating realistic faces and hands, as well as correcting the lines that are distorted due to inaccurate flow estimation of the warping module.

4.3. Learning model

For training, we use three loss functions, a perceptual loss $\mathcal{L}_{per}$, an affine preserving loss $\mathcal{L}_{aff}$, and a generative loss $\mathcal{L}_g$. In AlBahar et al. (2021), the authors propose to use an identity loss to preserve the identity of facial parts. However, our experiments showed that this could be misleading when the facial parts are occluded in either the source or the target images. In addition, Zhou et al. (2022) proposes to use a contextual loss to measure the similarity of non-aligned regions between the generated sample and the ground truth target image. However, this violates our goal to best fit the source image to the target pose without any non-aligned regions. $\mathcal{L}_{per}$ is utilized to ensure a pixel-level similarity between the generated image y_o and the target sample I_t, which is calculated as follows:

$$\mathcal{L}_{per}(y_o, I_t) = \sum \|\phi_i(y_o) - \phi_i(I_t)\|_1 \tag{6}$$

where $\phi_i(\S)$ is the i-th feature map of a pre-trained network (here we benefit from VGG-19 Simonyan & Zisserman (2014) pre-trained on Imagenet Deng et al. (2009)) when it is applied on $\S$. Inspired by Siarohin et al. (2019b), we use four different resolutions of y_o and I_t as input of our pre-trained model ϕ which ensures the similarity of

	Sz	UnWarpME	FOMM	MRAA	TPSMM	GFLA32	GFLA64
l_1-norm	32	**13.11**	13.26	13.79	13.66	14.18	14.01
l_1-norm	256	**15.52**	15.97	15.58	16.12	28.13	20.25
SSIM	256	0.608	0.604	0.607	**0.617**	0.554	0.615
LPIPS(Alex)	256	**0.39**	0.47	0.42	0.45	0.52	0.48
LPIPS(VGG)	256	**0.35**	0.42	0.38	0.41	0.49	0.44

Table 1: Comparison with the state-of-the-art on estimating the most accurate flow maps. The samples are generated by direct sampling from the source images and compared with the target samples.

the samples at different scales.

We also benefit from a generative loss, which encourages the photo-realism of the generated samples. To do so, both the warping and the refinement module are considered as the generator of our network $\mathcal{G} = \{\mathcal{W}, \mathcal{R}\}$ which competes against a discriminator $\mathcal{D}$. Our discriminator is conditioned on the target pose which means that both the generated and the target samples are first concatenated to the target pose and then passed to the discriminator. Our generative loss is defined as follows:

$$\mathcal{L}_g(y_o, I_t) = \mathbb{E}\big[\big(1 - \mathcal{D}(y_o, p_d)\big)\big] + \mathbb{E}\big[\mathcal{D}(I_t, p_d)\big] \tag{7}$$

where $y_o = \mathcal{G}(I_s, p_s, p_d)$.

Additionally, we use an affine preserving loss that enforces the network to preserve the linearity of the warping function in the regions where the target pose p_d is a linear transformation of the source pose.

$$\mathcal{L}_{aff}(p_t) = \|\mathcal{W}(p_{st}, p_t) - s_f\|_1 \tag{8}$$

where $\mathcal{W}(p_{st}, p_t)$ is the pixel-wise flow map estimated by the warping module. The overall loss function is defined as a weighted sum of $\mathcal{L}_{per}$, $\mathcal{L}_g$, and $\mathcal{L}_{aff}$, where λ_1 and λ_2 are empirically determined to ensure the best quality of the generated samples.

$$\mathcal{L}_t = \mathcal{L}_{per} + \lambda_1 \mathcal{L}_g + \lambda_2 \mathcal{L}_{aff} \tag{9}$$

Method	l_1-norm	SSIM	LPIPS(Alex)	LPIPS(VGG)
w/o $\mathcal{L}_{aff}$	42.13	0.504	0.84	0.92
Full	15.52	0.608	0.39	0.35

Table 2: Ablation study on $\mathcal{L}_{aff}$. We conduct experiments on the samples of size 256×256

Method	l_1-norm	SSIM	LPIPS(Alex)	LPIPS(VGG)
w/o Ref. Net.	42.13	0.504	0.84	0.92
Full	15.52	0.608	0.39	0.35

Table 3: Ablation study on the refinement module. We conduct two sets of experiments with and without the refinement module.

Method	l_1-norm	LPIPS(Alex)	LPIPS(VGG)
1-layer	33.31	0.61	0.58
2-layer	27.92	0.59	0.51
3-layer	21.17	0.47	0.44
4-layer	16.14	0.42	0.39
5-layer	15.96	0.40	0.35
6-layer	15.52	0.39	0.35

Table 4: Ablation study on the depth of the $\mathcal{C}_l$. It projects the 4096-channel Q_γ to a 2-channel flow map $f_{s,t}$

5. Experiments

This section evaluates the performance of our warping module on two real-world datasets: Deepfashion Liu et al. (2016) and Market-1501 Zheng et al. (2015). Deepfashion is a fashion-style dataset that includes 52,712 images, mostly captured indoors against a white background. Images are of the size 256×256, all provided in JPG format. We use the same split of data provided by Zhu et al. (2019), which includes 101,966 pairs of training samples and 8,570 test pairs. Each pair includes two images of the same person captured in different poses. The pose of each image is already extracted using OpenPose Cao et al. (2017).

5.1. Evaluation metrics

The evaluation is based on 3 different metrics: Structural Similarity Index Measure (SSIM) Wang et al. (2004), Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018), and l_1-norm.

5.2. Quantitative analysis

We compare our method with GFLA Ren et al. (2020), FOMM Siarohin et al. (2019b), TPSMM Zhao & Zhang (2022), and MRAA Siarohin et al. (2021). Similar to our method, they all provide an estimation of the warping flow in an unsupervised manner. FOMM, TPSMM, and MRAA propose to estimate the keypoints of RGB images using an internal module. However, for a fair comparison, we modify this strategy so that the flow map is directly estimated from two sets of given keypoints. We evaluate the performance of each method for estimating an accurate flow map, which is the main contribution of our method and the primary difference between these competing algorithms. To do this, we first estimate a flow map using each of these networks on a pair of source and target poses. The map is then rescaled to 256×256 pixels and utilized to warp the source image. Rescaling is performed using bilinear interpolation. Finally, the similarity between the warped image and the target sample is reported in terms of three different measures: SSIM, LPIPS, and l_1-norm.

The results are listed in Table 1. For convenience, we name our method UnWarpME, which is an acronym for Unsupervised Warping Map Estimation. The evaluation is provided at two different scales: 32×32 and 256×256. For the 32×32 scale, we first downsample each image to 32×32 and then upsample it to the original dimension. All the competing algorithms use the same split of the dataset, allowing for a fair comparison between the results. For training the competing methods, we follow the same learning rate and epoch number suggested in the original papers or official implementations.

As can be seen, there are two versions of GFLA at two different scales, 32×32 and 64×64, which refer to the two flow maps estimated by this method. These maps are separately estimated in the first stage of this method and individually contribute to two distinct layers of its second-stage network. However, for FOMM, TPSMM, and MRAA, they provide a set of individual flow maps, each estimated using a parametric model. These maps are later accumulated into a single flow map where each pixel of the final map is selected from one of these parametric estimations. From the table, our method outperforms the state-of-the-art on three out of four evaluation metrics and on two scales, demonstrating the superiority of our method in both reconstructing the global shapes of the target samples and generating textures in their correct positions.

5.3. Visualization

We further visualize a few samples of warped images generated by our method and each of the competing algorithms. The results are shown in Figure 2. As can be seen, none of the competing methods can introduce the invisible parts into the warped images (for additional results, please refer to Figures 7-10 at the end of the paper). GFLA suffers from unwanted distortions even when displacing visible patches to their new locations. This occurs because its flow estimator does not account for covering the invisible parts of the sample. In contrast, FOMM, TSPM, and MRAA can ideally displace the visible patches, but due to their parametric estimation, they fail to estimate a correct flow map between two complex poses. As a result, their attempts to introduce the invisible parts are limited to stretching or duplicating parts of the images. Unlike these strategies, our method can estimate a highly complex flow function that not only displaces the visible parts to their new locations but also introduces any novel parts indicated by the target pose, all by warping from the visible patches of the source sample.

5.4. Ablation study

We evaluate the efficiency of each block in our proposed algorithm. To do this, we consider two variants of our method: (1) our model without $\mathcal{L}_{aff}$, and (2) our model without the refinement network. All the experiments are conducted on the same split of the dataset as in Section 3.1.

The results are reported in Tables 2 and 3. Without $\mathcal{L}aff$, the model simply estimates the flow map using a pixel-wise transformation without any encouragement for preserving the vicinity of pixels. In this way, the model still generates the overall pose of the samples; however, it fails to generate a faithful appearance to the source images. The results in the tables validate this assumption, where the performance of the *model w/o $\mathcal{L}aff$* is significantly lower than the full model. This proves the effectiveness of $\mathcal{L}aff$ in preserving the fidelity of the textures, especially for the *visible* parts whose displacements are usually linear, even though $\mathcal{L}aff$ has no prior knowledge about the visibility of the pixels.

Without a refinement module, the model directly computes the flow map by minimizing the distance between the warped image and the target sample. In this way, all the background estimations are performed using the flow estimation module. This causes an overfitting issue when we fit too closely to the details of the background regions. This can be verified by the results in Table 3, where the refinement module significantly boosts the performance of our method on all metrics.

We also conduct another experiment to evaluate the performance of the model when using a projection network C_l with different numbers of layers. The experiment is performed with six different layers. The depth of the layers is respectively 128, 128, 96, 64, 32, and 2. As seen in Table 4, the best performance is achieved when we use $\mathcal{C}_l$ with five to six hidden layers, which are considered in the full model of our method.

5.5. Implementation details

We implement $\mathcal{M}$ as a fully convolutional network consisting of 11 convolutional layers, each followed by Batch Normalization Ioffe & Szegedy (2015). The last layer outputs a feature map $z_.$ with dimensions $35 \times 64 \times 64$, where 35 is the number of feature channels and 64×64 denotes the spatial size of the feature map. Each channel of this map is then normalized using l_2 normalization.

γ is implemented using Soft Mutual filtering Rocco et al. (2018), which retains $C(i, j, v, u)$ if $z_s(i, j)$ and $z_t(v, u)$ are the most similar pixels to each other when compared across both feature maps.

$$\gamma(i, j, v, u) = \frac{C(i, j, v, u)}{max_{\alpha\beta} C(\alpha, \beta, v, u)} \frac{C(i, j, v, u)}{max_{\alpha\beta} C(i, j, \alpha, \beta)} \tag{10}$$

where $max_{\alpha\beta}C(\alpha, \beta, v, u)$ denotes the maximum similarity between $z_t(v, w)$ and the entire pixels of $z_s(\alpha, \beta)$ when $\alpha \in \{1, ..., h_1\}$ and $\beta \in \{1, ..., w_1\}$.

For $\mathcal{C}_l$, we use a 6-layer convolutional network. The first layer receives a rasterized tensor of size $4096 \times 64 \times 64$, and the last layer outputs a flow map of size $2 \times 64 \times 64$. The depths of the layers are respectively 128, 128, 96, 64, 32, and 2. The encoder of the refinement network consists of two convolutional layers, each followed by a downsampling operation. The last layer outputs a feature map of size $256 \times 64 \times 64$. The decoder is implemented using 6 residual blocks with 2 upsampling layers. We apply Batch Normalization after each upsampling layer. The architecture of the residual blocks is depicted in Figure 4.

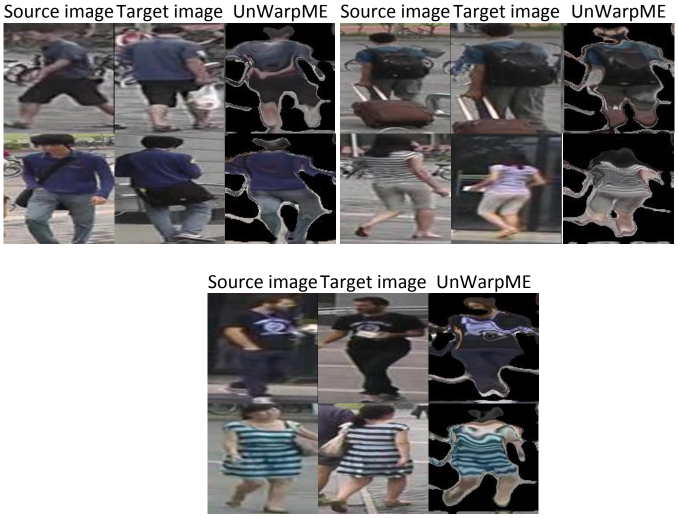

Figure 3: Generating a sample from the Market-1501 database by direct sampling from the source image.

5.6. Application for person re-identification

We also conducted another experiment to evaluate the performance of our method for recognition tasks. To do this, we trained our model on a paired-image dataset to learn a flow map that generates a novel view of a person. We then used this model to augment the training set of a re-identification database, increasing the number of poses that each person contributes to the training set. Note that we did not use the output of the Refinement network; instead, we generated the novel views by sampling the pixels using the estimated flow map. This approach allows us to evaluate the performance of our flow estimation model in a task that does not necessarily require detailed images for the recognition process.

Our experiment was conducted on the Market-1501 dataset, which contains 12,936 images of 751 identities as training samples and 19,732 images of 750 identities as queries. The images were collected outdoors using six different cameras.

For the flow field estimation, we used the paired dataset provided by Zhu et al. (2019). Note that our model is an algorithm that learns to estimate a flow map from two skeletal poses and is therefore not biased towards the appearance of the training samples. After training the flow estimator, we randomly selected 25 poses from the training set and transferred each of the training samples to all these novel poses. All background regions were replaced with gray. This process provided us with 323,400 training samples covering most of the natural poses a human takes while walking on the street. We then used this new dataset to fine-tune a pre-trained re-identification network and tested it on the query samples. We used DG-Net++ Zou et al. (2020) as our baseline model. The results are listed in Table 5. As can be seen, the results demonstrate the effectiveness of the augmentation on the performance of this method. Our method improves the mean Average Precision (mAP) and ranking accuracy at Rank 1, Rank 5, and Rank 10 compared to DG-Net++. Specifically, our approach achieves a higher mAP of 63.8% versus 61.7%, and improves Rank 1 accuracy from 82.1% to 83.8%, Rank 5 accuracy from 90.2% to 93.0%, and Rank 10 accuracy from 92.7% to 95.4%. The results indicate that a simple warping technique is sufficient to boost the performance of a re-identification task, eliminating the need for generating high-quality images, which is a more complex and time-consuming process. The reason for this boost is clear: augmentation generates diverse versions of the same image, providing more training examples. This increases the variations in captured poses of images, helping the model learn to recognize individuals under different conditions. This has already been verified by many other studies; however, in this work, we demonstrate that what is crucial for augmentation is augmenting the main subject in the scene and not the background. This can be easily achieved using a simple warping transformation rather than complex high-quality image generation. We also provide a visualization of our flow estimation technique on this dataset (Figure 3).

5.7. User study

We conducted a user study to evaluate the effectiveness of our proposed method based on human opinions. Six participants, all experienced in evaluating machine learning and computer vision tasks, were selected for the study. The evaluation focused on two specific questions: (1) Which algorithm better estimates the pose of the samples? (2) Which algorithm better generates the invisible parts of the samples?

Each participant viewed 80 different images, each accompanied by 6 samples generated by various warping algorithms, including our proposed method. The competing algorithms included FOMM, MRAA, TPSMM, GFLA 32, and GFLA 64. These images were randomly sampled from the test set of the DeepFashion database, as detailed in the introduction of Section 5. To familiarize the participants with the evaluation process, each participant initially viewed 17 images along with their novel syntheses. Subsequently, participants were asked to compare our method with each competing algorithm based on the aforementioned questions.

We employed a blind test strategy to mitigate bias in evaluations. This approach ensured participants were unaware of which method produced each image, thereby minimizing any preconceived notions during evaluation.

The overall results were determined by aggregating scores provided by all participants for the comparison between our method and each competing algorithm across all 80 images. The percentage of preferences for our method over the others is summarized in Table 6, indicating the consistent superiority of our method in sample generation in novel poses.

While the DeepFashion dataset provides comprehensive coverage of various human poses, it exhibits a bias towards human-centric images. As a result, while our findings can be generalized to real-world scenarios, they specifically apply to human images.

5.8. Skip connection

Skip connections have long been introduced in image-to-image translation networks to ensure the fidelity of generated samples to input images. This strategy has

	mAP	Rank1	Rank5	Rank10
DG-Net++	61.7	82.1	90.2	92.7
Our method+DG-Net++	63.8	83.8	93.0	95.4

Table 5: A detailed comparison between vanilla DG-Net++ and DG-Net++ when integrated by our method. The evaluation metrics include mean Average Precision (mAP) and ranking accuracy at Rank 1, Rank 5, and Rank 10.

Method	Q1	Q2
UnWarpME/FOMM	82.7/17.3%	90.6/9.4%
UnWarpME/MRAA	78.9/29.1%	84.7/15.3%
UnWarpME/TPSMM	80.8/19.2%	87.7/12.3%
UnWarpME/GFLA32	75.4/24.6%	77.2/22.8%
UnWarpME/GFLA64	70.8/29.2%	75.6/24.4%

Table 6: User study. The results are provided based on two different questions from 6 volunteers.

demonstrated effectiveness in early works such as Isola et al. (2017); Siarohin et al. (2018), and has been widely adopted by subsequent image-based translation approaches. In this section, we evaluate the impact of introducing a skip connection into our proposed strategy. Figure 4 illustrates the modified configuration of the left panel of Figure 1 after incorporating a skip connection into the first layer of the decoder.

In this configuration, the decoding process simultaneously samples from the convolutional feature map of the source image and its corresponding warped map. Our initial hypothesis was that this approach would enhance the preservation of fine-grained textures in the generated samples, akin to other image-to-image translation tasks. However, we observed that this method led to a failure in preserving pixel integrity, largely due to the Gradient Confusion (GC) effect Sankararaman et al. (2020). This effect arises from multiple sources of change for the same concept within the network.

Unlike traditional image-to-image translation tasks, where networks are typically encouraged to first produce a correct feature map before using it to generate the output image, skip connections allow the warped feature map to directly influence the generation of the output sample. This deviation complicates the learning process and undermines pixel integrity preservation. Figure 5 compares the resulting warped image produced

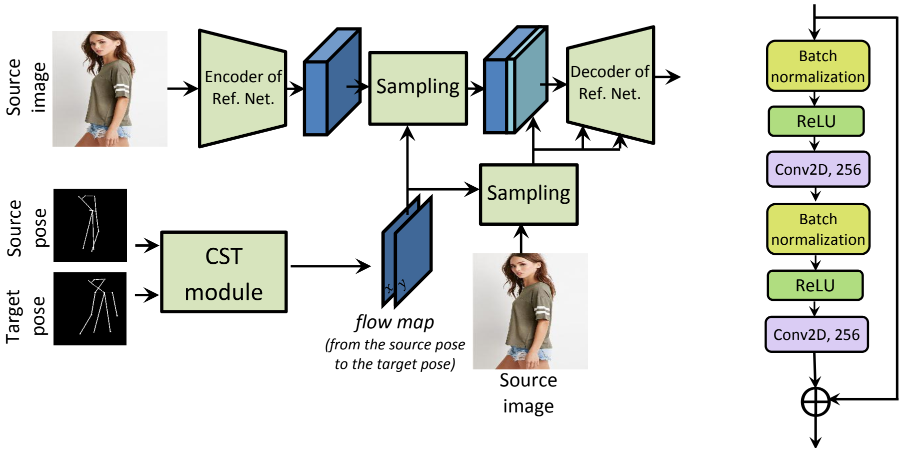

Figure 4: Left: Our model when using skip connection in the refinement module. Right: Residual Block of our refinement network.

with and without skip connections. As can be seen, skip connections cause some artifacts in the warped image, especially in the background region. This observation is consistent with our previous discussion about the ineffectiveness of skip connections in our method.

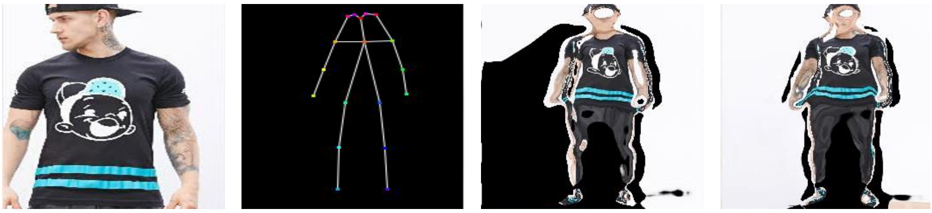

Figure 5: the effectiveness of skip connection, (a) source image, (b) target pose, (c) Generated sample using the flow map estimated by Fig 4 (there is a duplication of the source patches in the generated sample), (d) generated using the flow map of the configuration in Figure 1.

5.9. *A continuous discriminator*

Similar to the approach of Arjovsky & Bottou (2017), we used a continuous discriminator where real and fake samples are considered as two continuous distributions. This approach guarantees richer information and more nuanced gradients for the dis-

Method	LPIPS
Naive discriminator	0.39
Continuous discriminator	0.37

Table 7: LPIPS scores for the generated samples using different strategies for training the discriminator in our model

criminator. For this, we utilized a pre-trained encoder, specifically a VGG16 model, to project real images into an embedding space.

The experimental setup was similar to that in the introduction of Section 5, but the naive discriminator was replaced with a continuous one. We employ the LPIPS (Learned Perceptual Image Patch Similarity) metric to measure the difference in the quality of the generated samples. The results are presented in Table 7. As can be seen, using a continuous discriminator greatly improves the quality of the generated samples. However, it also significantly increases the computational cost of the discriminator.

6. Conclusion

We have introduced a novel warping module for spatial transformation of RGB images. Unlike existing affine-based transformations, our method utilizes a pixel-wise warping strategy that enables learning complex transformations using a single warping map. Comparisons with traditional affine-based methods have shown that our approach provides significant advantages in generating higher-quality transformations, accurately fitting an RGB image to its target pose.

Furthermore, we evaluated the performance of our warping strategy in augmenting re-identification tasks. We found that simply warping samples to new poses can substantially enhance the performance of re-identification tasks. This approach enables learning various poses of a specific person without the need to generate high-quality samples, which is time-consuming and requires large networks with excessive parameters.

Disadvantages Although our method effectively fits the source image to its new location in the target pose, it relies on a larger number of loss functions. Additionally, the refinement module in our method is divided into two basic parts. The first part is trained with a counter learning rate to learn the exact warping map used, while the second part is trained with a higher learning rate to produce photorealistic images.

Acknowledgment

To appear in camera ready version.

References

AlBahar, B., Lu, J., Yang, J., Shu, Z., Shechtman, E., & Huang, J.-B. (2021). Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. *ACM Transactions on Graphics (TOG)*, *40*, 1–11.

Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. ArXiv preprint arXiv:1701.04862.

Bello, J. L. G., & Kim, M. (2024). Novel view synthesis with view-dependent effects from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10413–10423).

Cao, A., Rockwell, C., & Johnson, J. (2022). Fwd: Real-time novel view synthesis with forward warping and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 15713–15724).

Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291–7299).

Corona, E., Pumarola, A., Alenya, G., Pons-Moll, G., & Moreno-Noguer, F. (2021). Smplicit: Topology-aware generative model for clothed people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11875–11885).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). IEEE.

Hasson, Y., Tekin, B., Bogo, F., Laptev, I., Pollefeys, M., & Schmid, C. (2020). Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 571–580).

Hou, Y., Solin, A., & Kannala, J. (2021). Novel view synthesis via depth-guided skip connections. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3119–3128).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* (pp. 448–456). PMLR.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).

Johnston, A., & Carneiro, G. (2019). Single view 3d point cloud reconstruction using novel view synthesis and self-supervised depth estimation. In *2019 Digital Image Computing: Techniques and Applications (DICTA)* (pp. 1–8). IEEE.

Li, Y., Huang, C., & Loy, C. C. (2019). Dense intrinsic appearance flow for human pose transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3693–3702).

Liu, M., Yan, X., Wang, C., & Wang, K. (2021). Segmentation mask-guided person image generation. *Applied Intelligence*, *51*, 1161–1176.

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1096–1104).

Luo, K., Wang, C., Liu, S., Fan, H., Wang, J., & Sun, J. (2021). Upflow: Upsampling pyramid for unsupervised optical flow learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1045–1054).

Lv, Z., Li, X., Li, X., Li, F., Lin, T., He, D., & Zuo, W. (2021). Learning semantic person image generation by region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10806–10815).

Minelli, G., Poggi, M., & Salti, S. (2023). Depth self-supervision for single image novel view synthesis. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5836–5843). IEEE.

Nguyen, H. C., Wang, T., Alvarez, J. M., & Liu, M. (2024). Mining supervision for dynamic regions in self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10446–10455).

Poddar, M., Mishra, A., Kewlani, M., & Pei, H. (2023). Self-supervised learning based depth estimation from monocular images. ArXiv preprint arXiv:2304.06966.

Ren, Y., Yu, X., Chen, J., Li, T. H., & Li, G. (2020). Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7690–7699).

Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., & Zha, H. (2017). Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Rocco, I., Arandjelovic, R., & Sivic, J. (2017). Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6148–6157).

Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., & Sivic, J. (2018). Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems*. volume 31.

Sabour, S., Tagliasacchi, A., Yazdani, S., Hinton, G., & Fleet, D. J. (2021). Unsupervised part representation by flow capsules. In *International Conference on Machine Learning* (pp. 9213–9223). PMLR.

Sankararaman, K. A., De, S., Xu, Z., Huang, W. R., & Goldstein, T. (2020). The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *International Conference on Machine Learning* (pp. 8469–8479). PMLR.

Sarkar, K., Golyanik, V., Liu, L., & Theobalt, C. (2021). Style and pose control for image synthesis of humans from a single monocular view. *arXiv preprint arXiv:2102.11263*, .

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., & Sebe, N. (2019a). Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2377–2386).

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., & Sebe, N. (2019b). First order motion model for image animation. *Advances in Neural Information Processing Systems*, *32*, 7137–7147.

Siarohin, A., Sangineto, E., Lathuiliere, S., & Sebe, N. (2018). Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3408–3416).

Siarohin, A., Woodford, O. J., Ren, J., Chai, M., & Tulyakov, S. (2021). Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13653–13662).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:1409.1556.

Stone, A., Maurer, D., Ayvaci, A., Angelova, A., & Jonschkowski, R. (2021). Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3887–3896).

Tabejamaat, M., Negin, F., & Bremond, F. (2024). Improving texture integrity through second-order constraints on warping maps. *Neurocomputing*, *591*, 127739.

Tabejamaat, M., Negin, F., & Bremond, F. F. (2021). Guided flow field estimation by generating independent patches. In *BMVC 2021-32nd British Machine Vision Conference*.

Tang, H., Bai, S., Torr, P. H., & Sebe, N. (2020). Bipartite graph reasoning gans for person image generation. *arXiv preprint arXiv:2008.04381*, .

Tang, H., & Sebe, N. (2021). Total generate: Cycle in cycle generative adversarial networks for generating human faces, hands, bodies, and natural scenes. *arXiv preprint arXiv:2106.10876*, .

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2387–2395).

Truong, P., Danelljan, M., Gool, L. V., & Timofte, R. (2020). Gocor: Bringing globally optimized correspondence volumes into your neural network. In *Advances in Neural Information Processing Systems* (pp. 14278–14290). volume 33.

Wang, Y., Liang, Y., Xu, H., Jiao, S., & Yu, H. (2024). Sqldepth: Generalizable self-supervised fine-structured monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 5713–5721).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*, 600–612.

Yang, X., Ma, Z., Ji, Z., & Ren, Z. (2023). Gedepth: Ground embedding for monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 12719–12727).

Yoon, J. S., Kim, K., Gallo, O., Park, H. S., & Kautz, J. (2020). Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5336–5345).

Zhang, J., Siarohin, A., Tang, H., Chen, J., Sangineto, E., Wang, W., & Sebe, N. (2021). Controllable person image synthesis with spatially-adaptive warped normalization. *arXiv preprint arXiv:2105.14739*, .

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 586–595).

Zhao, J., & Zhang, H. (2022). Thin-plate spline motion model for image animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3657–3666).

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1116–1124).

Zhou, X., Yin, M., Chen, X., Sun, L., Gao, C., & Li, Q. (2022). Cross attention based style distribution for controllable person image synthesis. In *European Conference on Computer Vision* (pp. 161–178). Springer.

Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., & Bai, X. (2019). Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2347–2356).

Zou, Y., Yang, X., Yu, Z., Kumar, B. V. K., & Kautz, J. (2020). Joint disentangling and adaptation for cross-domain person re-identification. In *European Conference on Computer Vision* (pp. 87–104). Springer.

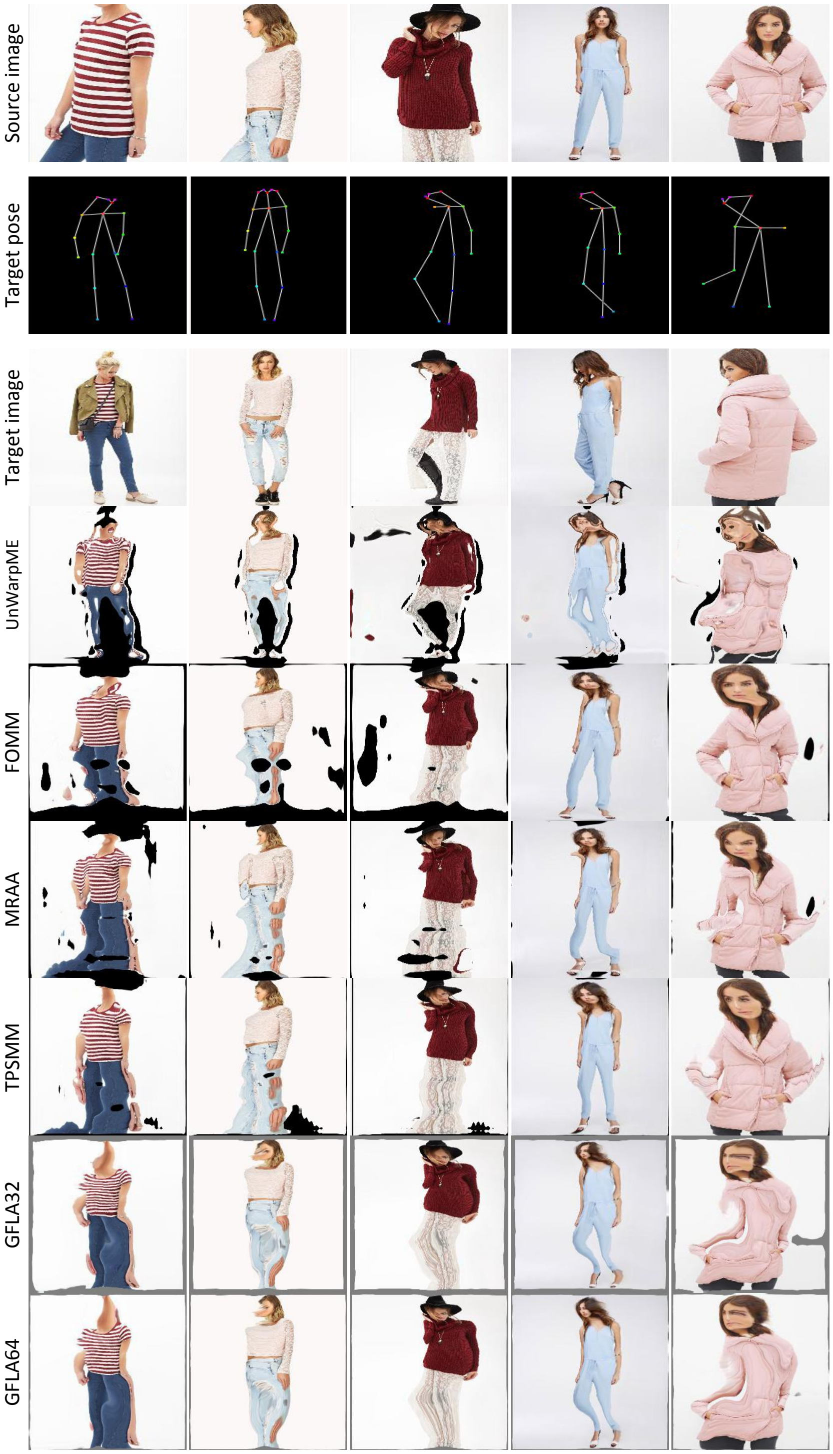

Figure 6: Additional comparison between our method and the state-of-the-art.

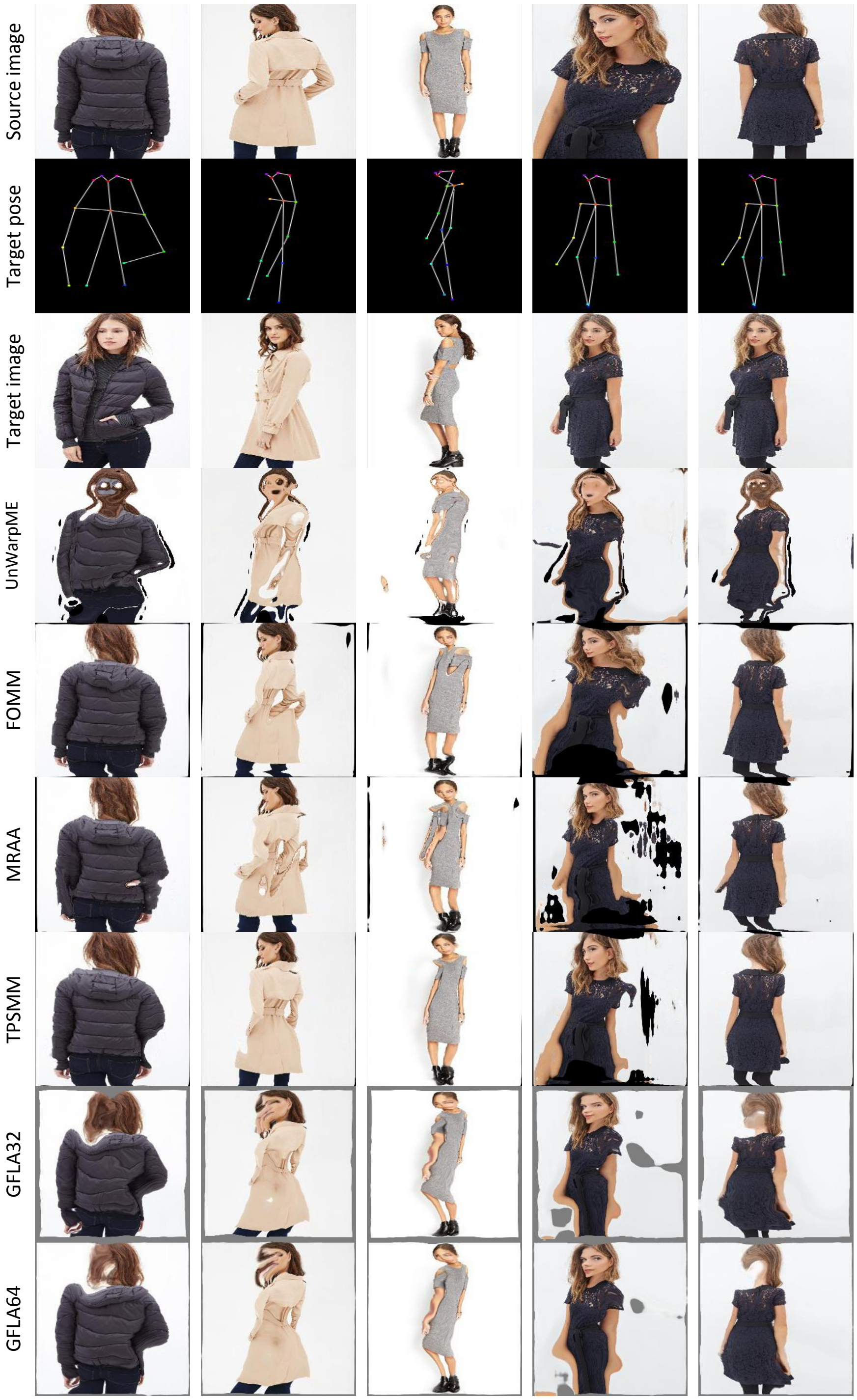

Figure 7: Additional comparison between our method and the state-of-the-art.

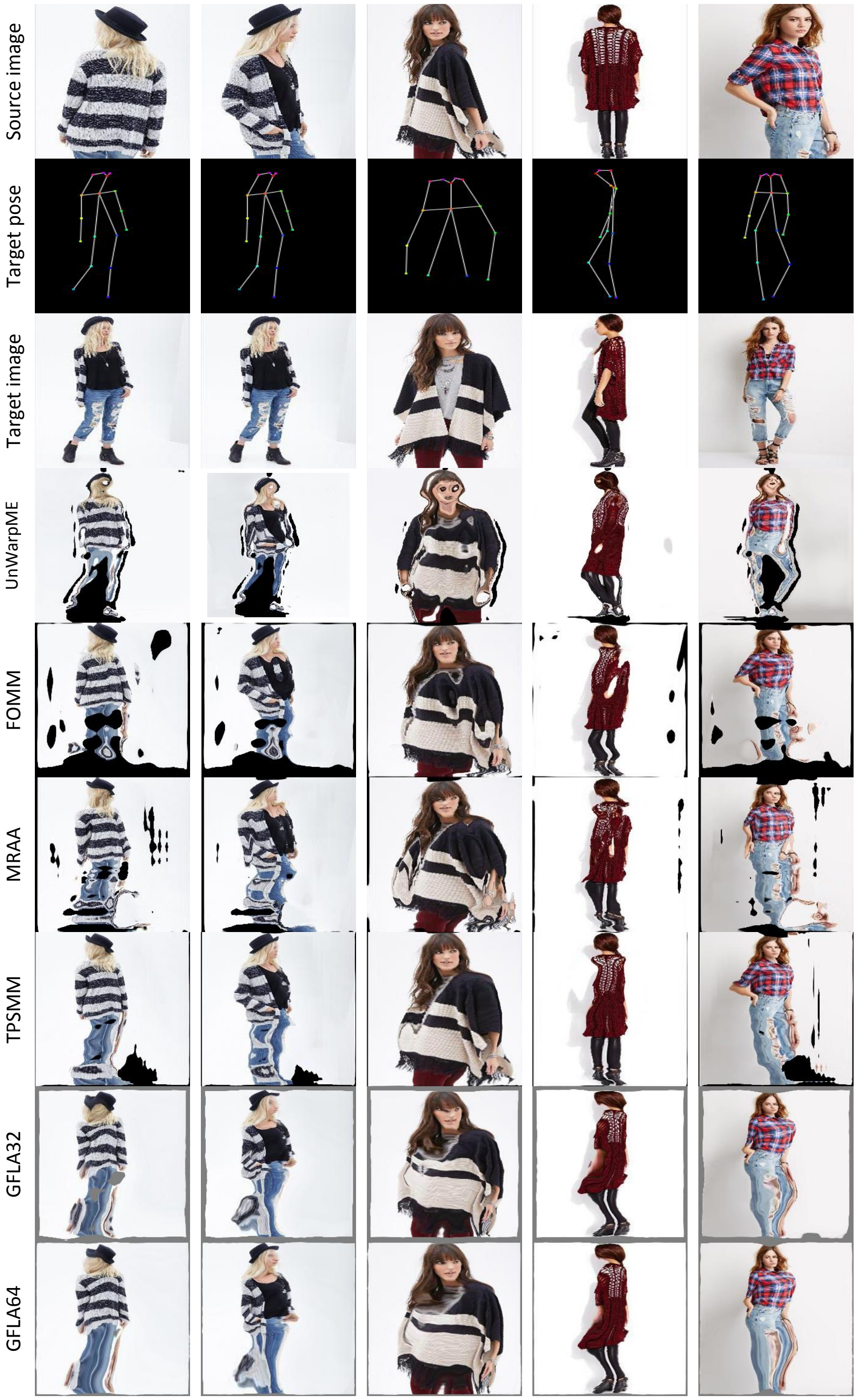

Figure 8: Additional comparison between our method and the state-of-the-art.

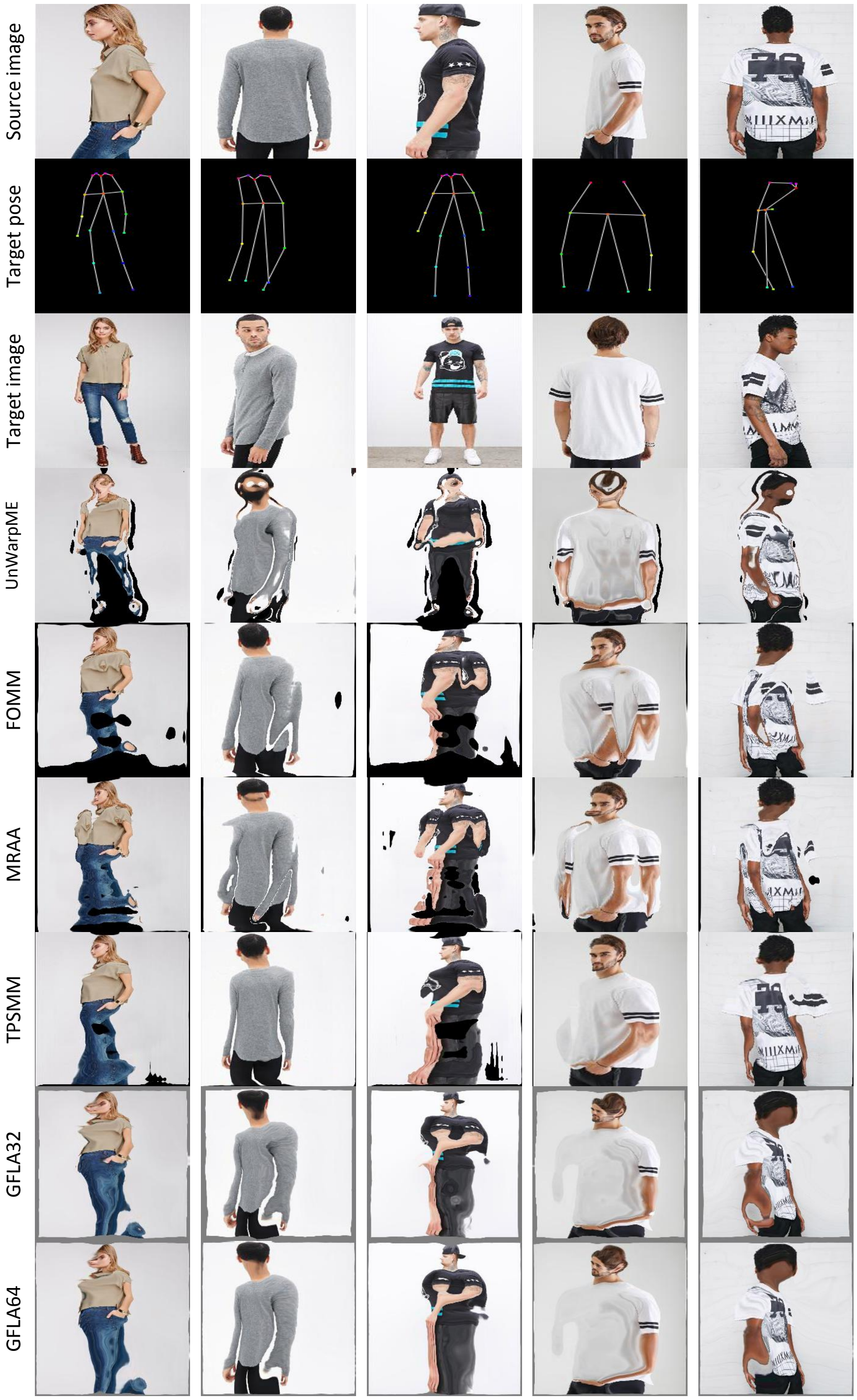

Figure 9: Additional comparison between our method and the state-of-the-art.