Model-Aware Gesture-to-Gesture Translation

Hezhen Hu [1], Weilun Wang [1]*, Wengang Zhou [1,2]†, Weichao Zhao [1], Houqiang Li [1,2]†
[1] CAS Key Laboratory of GIPAS, EEIS Department, University of Science and Technology of China (USTC)
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{alexhu, wwlustc, saruka}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn

Abstract

Hand gesture-to-gesture translation is a significant and interesting problem, which serves as a key role in many applications, such as sign language production. This task involves fine-grained structure understanding of the mapping between the source and target gestures. Current works follow a data-driven paradigm based on sparse 2D joint representation. However, given the insufficient representation capability of 2D joints, this paradigm easily leads to blurry generation results with incorrect structure. In this paper, we propose a novel model-aware gesture-to-gesture translation framework, which introduces hand prior with hand meshes as the intermediate representation. To take full advantage of the structured hand model, we first build a dense topology map aligning the image plane with the encoded embedding of the visible hand mesh. Then, a transformation flow is calculated based on the correspondence of the source and target topology map. During the generation stage, we inject the topology information into generation streams by modulating the activations in a spatially-adaptive manner. Further, we incorporate the source local characteristic to enhance the translated gesture image according to the transformation flow. Extensive experiments on two benchmark datasets have demonstrated that our method achieves new state-of-the-art performance.

1. Introduction

Hand gesture-to-gesture translation aims to convert the source gesture to the target one conditioned by the target posture, while preserving identity information. This problem is of significant importance with broad applications in sign language production, data augmentation, human-computer interactions, *etc*. Hand exhibits highly articulated joints and covers almost uniform appearance with fewer local characteristics compared with rigid human bodies or faces. As a result, it is characterized by more fine-grained texture with self-occlusion and poses a new challenge on learning the precise correspondence between the source and the target for the task of gesture-to-gesture translation.

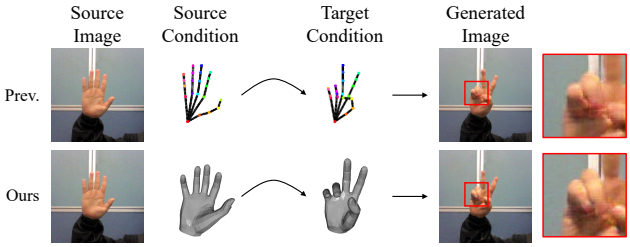

Figure 1. Illustration of gesture-to-gesture translation. The first and second row visualize the intermediate condition and generated image of the previous method [34] and our method, respectively. The samples are chosen from the STB dataset. Our method exhibits more accurate spatial structure and fine-grained details.

Previous methods [34, 19] encode the gesture state via 2D sparse joint representation. The source hand images are translated according to the optical flow learned from the source and target 2D joints. GestureGAN [34] attempts to learn the mapping under two novel losses in a cycle-consistency manner. As shown in Fig. 1, we visualize the intermediate representation and generated image by previous work and our method. It can be observed that the generated image by the comparison method contains incorrect and blurry structure, while our method produces the image with more fidelity in fine-grained details.

Previous methods follow a direct data-driven paradigm and suffer unsatisfactory results due to limited representation capability and intrinsic ambiguity of 2D sparse hand pose [34, 26]. To tackle this issue, we incorporate hand prior and propose a model-aware gesture-to-gesture translation framework. Since hand gesture-to-gesture translation needs fine-grained understanding of hand pose, shape and texture, we attempt to depict the hand status more informatively. Specifically, we extract the hand representation in a model-aware way. The hand model provides a compact mapping from the latent pose and shape embedding to the high-dimensional hand mesh representation. It is a fully-

*Contribute equally with the first author.
†Corresponding author: Wengang Zhou and Houqiang Li.

differentiable statistical model, which stores prior knowledge learned from a large variety of hand scans. With this model, the hand is reconstructed with more details, while irrational hand poses are filtered out. During the extraction process, we only need to estimate the latent embedding and camera parameter matching the image. To this end, our method provides two alternative effective ways, including direct regressing and iterative fitting.

To fully exploit structure information contained in the hand mesh representation, we first build the dense topology map for the source and target, and transformation flow between them. Specifically, we unravel the surface of the hand model and create its flattened representation in 2D space. Then each hand mesh face visible in the aligned image plane is encoded with its corresponding position embedding from the flattened surface representation. The transformation flow is derived by calculating the correspondence between the source and target. In this way, the dense topology map and their transformation flow preserve abundant structure information for the next stage. When turning to the gesture synthesis stage, we modulate the structure into generation streams in a spatially-adaptive manner. To further enhance the translated hand gesture, we adaptively incorporate local characteristics with the attention mechanism.

Our contributions are summarized as follows,

- To our best knowledge, we propose the first model-aware gesture-to-gesture translation framework, consisting of three key modules, *i.e.,* hand representation extraction, hand topology modeling and gesture synthesis.
- We introduce hand prior with hand meshes as the intermediate representation, and propose an alternative hand representation extraction method based on iterative fitting besides the direct regressing way.
- Extensive experiments on two widely-used benchmarks, *i.e.,* STB and Senz3D, demonstrate the effectiveness of our proposed method, achieving new state-of-the-art performance. Our generated gesture images have more accurate spatial structure with better fine-grained details.

2. Related Work

In this work, we will briefly review the related topics, including pose-guided image translation, gesture-to-gesture translation and hand representation.

2.1. Pose-Guided Image Generation

Pose-guided image generation aims to generate an image by combining the appearance of an object in the source image and the target pose in the target image. Existing methods on pose-guided image generation mainly focus on the human image generation [29, 18, 5, 26]. DeformableGAN [29] introduces deformable skip connections into the generator to move local information according to human structural deformations. VU-Net [6] employs a U-Net [28] to generate person images conditioned on the appearance vector encoded by a variational autoencoder [16]. Ren *et al.* [26] propose a global-flow local-attention framework to generate vivid clothes textures for targets. Gesture-to-gesture translation is also a pose-guided image generation problem in which the output image is generated by warping the source hand image while the gesture of the output image is conditioned by the target hand posture.

2.2. Gesture-to-Gesture Translation

In recent years, more and more researchers have turned their attention to gesture-to-gesture translation. Several methods [19, 34] have been proposed to translate the hands from one gesture to another based on hand keypoints, skeleton, *etc*. For instance, Liu *et al.* propose a generative adversarial network, GestureGAN [34], with a novel color loss for high-quality results. Tang *et al.* [19] introduce a Δ-GAN, which performs this task in the wild with a simple-to-draw annotation. However, the above methods represent pose with only 2D keypoints, which often leads to artifacts and unreasonable finger configuration in the final generated image. It is partially attributed to the limited representation capability of 2D keypoints. Complex hand posture with self-occlusion may cause ambiguous expression of pose, which will result in blurry or incorrect translation results.

2.3. Hand Representation

Hand plays an important role in the human-centric video understanding, where recovering hand poses and shapes from images enables many real-world applications, including data augmentation, virtual reality, *etc.* There exist many methods focusing on sparse hand poses, *i.e.,* 2D or 3D skeletal articulations [30, 3, 40, 11, 36, 2]. Compared with sparse representation, modeling hand densely is able to express more information. Previous models utilize various techniques, including modeling the shape primitive [23], sum-of-Gaussians [32], sphere-mesh [35], *etc*. However, these models only roughly approximate the hand shape with artifacts. Further, a triangulated mesh with linear blend skinning (LBS) is adopted to make the model more realistic. Da La Gorce *et al.* [4] propose a triangulated hand model with the scaling terms for each bone to change hand shape. MANO [27] is the most popular hand model, and applied to hand tracking [9], hand pose estimation [1], *etc.* It is a fully-differentiable statistical model, learned from a large variety of hand scans. Considering the capability of MANO to deform the hand mesh with the pose and learned pose-dependent shape corrections, in this work, we adopt it to depict the gesture state with incorporated hand prior.

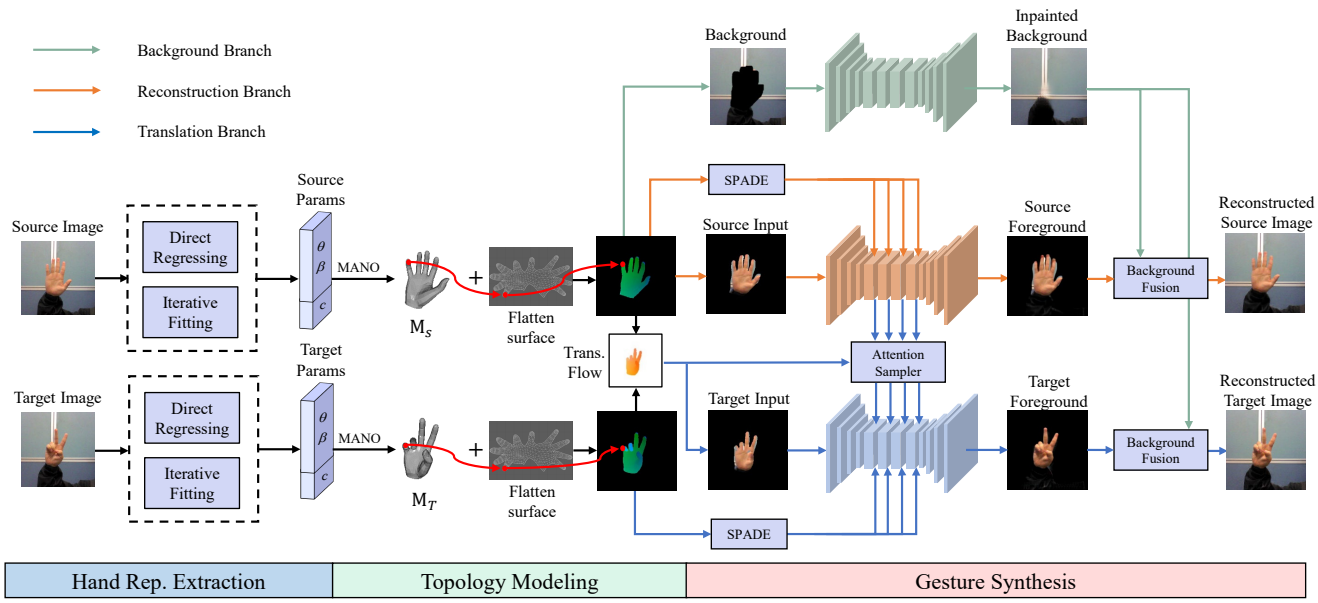

Figure 2. The overview of our framework, which contains three components: *Hand Representation Extraction*, *Topology Modeling* and *Hand Gesture Synthesis*. *Hand Representation Extraction* depicts the gesture state with expressive model-aware representation. *Topology Modeling* encodes the hand topology aligning the 2D image plane and calculates the transformation flow between the source and target. *Hand Gesture Synthesis* further takes advantage of the structure information and generates the reconstructed target image.

3. Our Approach

In this section, we first give an overview of our framework. Then we elaborate each component and loss function of the framework. Under our task definition, our framework generates hand images conditioned by target poses which maintain realistic appearance and fine-grained textures of the source hand. As shown in Fig. 2, our framework introduces hand prior and depicts the hand gesture status with more expressive high-dimensional model-aware representation, jointly with the estimated camera parameter. Then we unravel the hand surface to the flattened 2D space and encode the visible hand mesh face with the position embedding in this flattened space. Through this way, we are able to get the dense hand topology map and further calculate the transformation flow between them. Finally, these intermediate representations are fed into the generation streams to generate the reconstructed target image.

3.1. Hand Representation Extraction

MANO hand model. Considering the limited representation capability of 2D pose, we choose to represent the hand more densely with hand prior incorporated. Specifically, we choose the fully differentiable MANO to generate the model-aware hand representation. MANO [27] provides a compact mapping from the low-dimensional pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$ to the triangulated hand mesh $\mathbf{M} \in \mathbb{R}^{N_v \times 3}$ with N_v =778 vertices and N_f =1538 faces. Specifically, the pose and shape are constrained in a lower dimensional PCA space to produce a physically plausible mesh. The PCA space is learned from a large dataset of 3D articulated hand scan. The mapping function is formulated as follows,

$$\mathbf{M}(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(\mathbf{T}(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}), \tag{1}$$

$$\mathbf{T}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \bar{\mathbf{T}} + B_S(\boldsymbol{\beta}) + B_P(\boldsymbol{\theta}), \tag{2}$$

where $B_S(\cdot)$ and $B_P(\cdot)$ denote shape and pose blend functions, respectively. $\mathbf{W}$ is a set of blend weights. Based on the pose and shape corrective blend shapes, *i.e.,* $B_P(\boldsymbol{\theta})$ and $B_S(\boldsymbol{\beta})$, the hand template $\bar{\mathbf{T}}$ is posed and skinned. Further, by rotating each part around joints $J(\boldsymbol{\beta})$ using the linear skinning function $W(\cdot)$ [15], the output mesh is generated. Besides the hand mesh, a more compact 3D representation $\widetilde{J}_{3D}$ can also be derived by the relevant vertices, where the original MANO model provides 16 3D joints. To keep consistent with the widely-used OpenPose annotation, we further add 5 extra vertices as the fingertips. Thus totally 21 3D joints are derived.

To make the model-aware hand representation match the given RGB image, the MANO input, *i.e.,* $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, along with the camera parameter, needs to be accurately estimated. To this end, we introduce the following two methods, *i.e.,* direct regressing and iterative fitting method.

Direct CNN regressing. We adopt the widely used framework [1] with a few modifications. The RGB gesture frame is fed into the ResNet34 [10] to generate a high-dimensional semantic representation, followed by a fully-connected layer to directly regress the latent embedding, *i.e.,* $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, and the camera parameter $\boldsymbol{c}$ for projecting the mesh to the 2D image plane.

Iterative model fitting. Besides the direct regressing method, latent embedding can also be derived from iterative fitting. Specifically, we fit the model to 2D hand joints by minimizing the following objective function,

$$E(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{c}) = E_j + \lambda_l E_l + \lambda_r E_r, \tag{3}$$

where E_j, E_l and E_r denote the joint, link and regularization term, respectively. For the joint term E_j, it attempts to minimize the distance between the 2D joints as follows,

$$E_j = \sum_i \mu_i \omega_i G(\Pi_{\boldsymbol{c}}(\widetilde{J}_{3D})_i) - J_i), \tag{4}$$

where $\Pi_{\boldsymbol{c}}(\cdot)$ denotes the projection function based on the camera parameter $\boldsymbol{c}$ and J denotes the 2D hand joints from manual annotation or off-the-shelf 2D joint extractor as supervision. To reduce the influence of noisy label, we adopt the GeMan-McClure penalty function $G(\cdot)$ [7] with weighting parameter (μ_i and ω_i for predefined per-joint and label confidence weight, respectively).

Since the joint-level constraint may ignore the hand structure, we further add a link term as follows,

$$E_l = \sum_{i,j} ||(\Pi_{\boldsymbol{c}}(\widetilde{J}_{3D})_i) - \Pi_{\boldsymbol{c}}(\widetilde{J}_{3D})_j)| - |J_i - J_j||. \tag{5}$$

Besides, to make the model produce a plausible mesh, a regularization term is added as follows,

$$E_r = \|\boldsymbol{\theta}\|_2^2 + w_\beta \|\boldsymbol{\beta}\|_2^2, \tag{6}$$

where w_β denotes the weighting factor.

3.2. Topology Modeling

Given the model-aware hand representation, we first build dense hand topology depicting the hand status aligning to the 2D image plane. Specifically, we first unravel the hand surface and construct the mapping from the mesh face to the flattened 2D space, *i.e.,* $\mathbf{F} \in \mathbb{R}^{N_f \times 2}$ [22]. The same mesh face will have the same mapping representation, ignoring what posture the hand mesh represents, and vice versa. Then, the model-aware hand representation is rendered based on the estimated camera parameter $\boldsymbol{c}$ as follows,

$$\{\mathbf{S}, \mathbf{V}\} = R(\mathbf{M}|\boldsymbol{c}), \tag{7}$$

where $R(\cdot)$ denotes the rendering function, $\mathbf{S}$ and $\mathbf{V}$ denote the binary hand silhouette mask and visible hand mesh index map aligning the 2D RGB image. For the area corresponding to the hand region, we index each visible mesh face to the flattened space, and encode each face with its position embedding in this space as follows,

$$\mathbf{O}(i,j) = \begin{cases} \vec{\mathbf{0}}, & \mathbf{S}(i,j) = 0, \\ \mathbf{F}(\mathbf{V}(i,j)), & \mathbf{S}(i,j) = 1, \end{cases} \tag{8}$$

where $\mathbf{O}$ denotes the aligned dense hand topology map with the same size as the original RGB image. The generated topology contains kind of geometric continuity: parts belonging to the same finger will have relatively close distance in the flattened space. Given topology maps between the source and target, the transformation flow $\mathbf{T}$ is derived by calculating their correspondence.

3.3. Hand Gesture Synthesis

We design an image generation network aware of the fine-grained hand structure, as shown in Fig. 2. It consists of three branches: *Background Branch*, *Reconstruction Branch* and *Translation Branch*.

The *Background Branch* inpaints the background cropped from the source image. The *Reconstruction Branch* does not directly generate the component of the final output. Instead, it takes the hand image cropped from the source image as input and reconstructs the source image cooperating with the *Background Branch*. In our network, the *Reconstruction Branch* works as an autoencoder. It injects the features of the middle layer into the *Translation Branch* to assist the generation of the final output in an attention-sampling manner. In the *Translation Branch*, we first warp the source image with the transformation flow obtained from the topology modeling stage and synthesize a rough image of the transferred foreground. The rough image is fed into a U-Net based generator [28]. Then the generator fuses the features from the *Reconstruction Branch* and finally generates the refined hand foreground. The generated foreground and background of the transfer result are merged in the *Background Fusion* module to generate the final hand gesture image.

To take full advantage of structure information contained in hand representation, we adopt the spatially-adaptive normalization (SPADE) [24] to make the network aware of the inserted the structure information. The SPADE injects the condition information into the generation streams by modulating the activations with a spatially-adaptive, learned transformation, which is formulated as follows,

$$h^{i+1}_{c,y,x} = \gamma^i_{c,y,x}(\mathbf{m}) \frac{h^i_{c,y,x} - \mu^i_c}{\sigma^i_c} + \beta^i_{c,y,x}(\mathbf{m}), \tag{9}$$

where $h^i_{c,y,x}$ and $h^{i+1}_{c,y,x}$ are the input and output activation of SPADE. μ^i_c and σ^i_c are the mean and standard deviation of the input activation. $\gamma^i_{c,y,x}(\mathbf{m})$ and $\beta^i_{c,y,x}(\mathbf{m})$ are the spatially-adaptive transformation learned from the input conditional map $\mathbf{m}$.

In the SPADE structure, we take the aforementioned topology map $\mathbf{O}$ obtained from the second stage as the conditional map $\mathbf{m}$. As mentioned above, the topology map encodes the visible hand mesh face with position embedding in the flattened surface space. It presents the structure

and boundary information, which is significant to generate structured texture with fine-grained details.

For generation of the target image, the features from the source image are warped according to transformation flow to align with the target hand gesture when fused with the *Translation Branch*. Some previous methods [13, 29] sample at a single point, which ignores contextual information and often leads to blurry and inaccurate results. Thus, we attempt to sample from a local patch in an adaptive, learnable manner by attention-based sampling [26]. The attention-based sampling is described as follows,

$$f_{attn}^{l} = \sum_{u} \sum_{v} \mathbf{K}^{l}(u,v) \mathbf{N}_{\mathbf{S}}^{l+\mathbf{T}_l}(u,v), \tag{10}$$

where the output activation value at the position l is denoted as f_{attn}^{l}, and $\mathbf{N}_{\mathbf{S}}^{l+\mathbf{T}_l}$ is the patch extracted from the source features at the position $l + \mathbf{T}_l$ according to the transformation flow $\mathbf{T}_l$, which is weighted by a learnable kernel $\mathbf{K}^{l}$.

3.4. Loss Function

Our framework is optimized under the weighted summation of multiple objective functions, *i.e.,* reconstruction loss, adversarial loss and regularization loss.

Reconstruction Loss. Since the generation stage needs to reconstruct both the source and target image, we employ two reconstruction loss terms. For reconstruction of the source image, we employ the L_1 loss as follows,

$$\mathcal{L}_{rec}^{src} = \|x_s - \hat{x}_s\|_1, \tag{11}$$

where x_s and $\hat{x}_s$ refer to the ground-truth and reconstructed source image, respectively. For the reconstruction of the target image, we employ the perceptual loss [14] as follows,

$$\mathcal{L}_{rec}^{tgt} = \sum_{i} \|f_i(x_t) - f_i(\hat{x}_t)\|_1, \tag{12}$$

where x_t and $\hat{x}_t$ refer to the ground-truth image and the reconstructed target image, respectively. $f_i(\cdot)$ is the i-th layer's feature maps extracted from a pre-trained VGG network [31]. In summary, the overall reconstruction loss is formulated as follows,

$$\mathcal{L}_{rec} = \mathcal{L}_{rec}^{tgt} + \lambda_{rec}\mathcal{L}_{rec}^{src}, \tag{13}$$

where the weight λ_{rec} is set to 1 in our experiment.

Adversarial Loss. We develop the adversarial learning [8] on the synthesized images. The adversarial loss regularizes the distribution of the generated images to that of the ground-truth images, which promotes the visual effect of generated images. For the generator $G(\cdot)$ and the discriminator $D(\cdot)$, we use the LSGAN$_{-110}$ [20] loss, which is formulated as follows,

$$\mathcal{L}_{adv}^{D} = \mathbb{E}_{x_r}[(1 - D(x_r))^2] + \mathbb{E}_{x_f}[(1 + D(x_f))^2], \tag{14}$$

$$\mathcal{L}_{adv}^{G} = \mathbb{E}_{x_f}[D(x_f)^2], \tag{15}$$

where x_r indicates the real image distribution while x_f indicates the generated image distribution. Meanwhile, we employ the patch-wise discriminator inspired by [12] in our adversarial learning.

Regularization Loss. It regularizes the generated foreground mask $\hat{\mathbf{S}}$ used in the *Background Fusion* module to be smooth and roughly equivalent to the original mask $\mathbf{S}$ obtained from the *Topology Modeling* stage. The regularization on the mask $\hat{\mathbf{S}}$ smoothness is formulated as follows,

$$\mathcal{L}_{reg}^{smth} = \frac{1}{HW} \sum_{i} \sum_{j} (\|\nabla_x \hat{\mathbf{S}}_{ij}\|_1 + \|\nabla_y \hat{\mathbf{S}}_{ij}\|_1). \tag{16}$$

The BCE regularization loss is utilized to ensure the rough equivalence between $\hat{\mathbf{S}}$ and $\mathbf{S}$, which is formulated as follows,

$$\mathcal{L}_{reg}^{bce} = \frac{1}{HW} \sum_{i} \sum_{j} (\mathbf{S}_{ij} log(\hat{\mathbf{S}}_{ij}) + (1 - \mathbf{S}_{ij}) log(1 - \hat{\mathbf{S}}_{ij})). \tag{17}$$

Finally, overall regularization loss is formulated as follows:

$$\mathcal{L}_{reg} = \mathcal{L}_{reg}^{bce} + \lambda_{reg}\mathcal{L}_{reg}^{smth}, \tag{18}$$

where the weight λ_{reg} is set to 1 in our experiment.

Overall Loss. The final objective function is the weighted summation of these loss terms as follows,

$$\mathcal{L}_{\mathcal{G}} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{adv}^{G} + \lambda_3 \mathcal{L}_{reg}, \tag{19}$$

where λ_1, λ_2 and λ_3 are the weighting factors to balance different types of losses. In our experiment, λ_1, λ_2 and λ_3 are set to 10, 10, and 1, respectively.

4. Experiments

4.1. Experiment Setup

Datasets. To evaluate the effectiveness of our method, we train and test our framework on two popular datasets, *i.e.,* STB [37] and Senz3D [21]. The STB dataset contains 18,000 images each recording a person performing various dynamic gestures under various backgrounds. It also provides the ground-truth 2D and 3D hand joint annotation. Senz3D [21] includes static gestures from 4 people. Each person performs 11 various static gestures 30 times in the frontal view of a Creative Senz3D camera, containing a total of 1320 images. Since it contains no annotation on the hand pose, we use the OpenPose [30] as the extractor. In both datasets, all images are in the same resolution of 640×480.

Implementation Details. For iterative model fitting, we utilize the L-BFGS optimizer with the Wolfe line search.

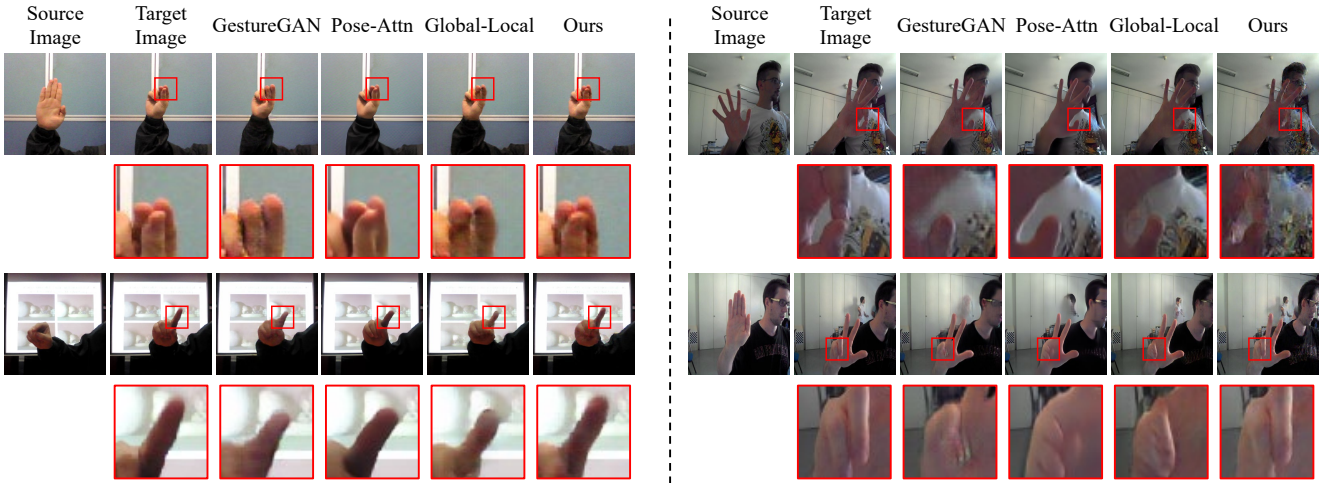

Figure 3. Qualitative comparisons with several state-of-the-art methods including GestureGAN [34], Pose-Attn [39] and Global-Local [26] on the STB (the left) and Senz3D (the right) datasets. Compared with our method, the images generated by these previous methods show inferior performance on the spatial structure and fine-grained details.

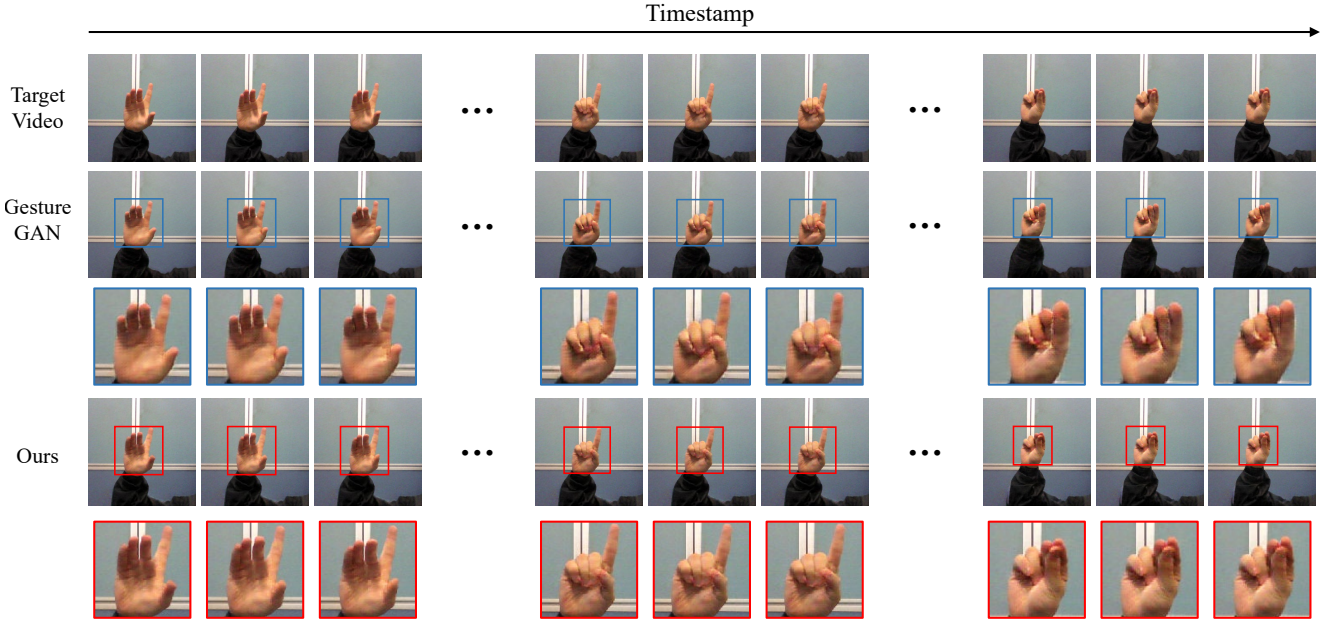

Figure 4. Qualitative results of translating the same source image to a target pose sequence in a frame-by-frame manner. We visualize 3 clips from one generated video sequence. In each clip, 3 continuous frames are extracted. It can be observed that images generated by GestureGAN [34] change *intensely* in detail when target poses change slightly. In contrast, the images generated by our method are *coherent* in detail under this circumstance.

The optimization follows a multi-staged approach similar to [25]. We first estimate the camera parameter. Then we optimize the input of hand model by gradually lowering weight for regularization term and increasing weight for joint and link terms. For the whole framework, we utilize the Adam optimizer. The training lasts 30 epochs in total. The learning rate starts at 2e-4 and linearly reduces after 5 epochs. The whole framework is implemented on PyTorch and we perform experiments on NVIDIA RTX TITAN.

For the STB dataset, since the hand only covers a relatively small region, we crop the hand region based on the 2.2 times width of the least square surrounding the 2D annotation. Then we resize the cropped image to 256×256. For the Senz3D dataset, since the hand has a dominant size, we directly resize the original image to 256×256.

Evaluation Metrics. We design an evaluation protocol to indicate the performance of different gesture-to-gesture translation methods, including both quantitative and qualitative evaluation. For quantitative evaluation, Structural Similarity Index Measure (SSIM), Inception Score (IS) and Learned Perceptual Similarity (LPIPS) [38] are employed to evaluate the generated images. Given the generated im-

Methods	STB			Senz3D		
	SSIM ↑	IS ↑	LPIPS ↓	SSIM ↑	IS ↑	LPIPS ↓
GestureGAN	0.9979	2.45±0.04	0.0939	0.9876	5.55±0.27	0.2042
Pose-Attn	0.9978	2.24±0.08	0.1360	0.9830	6.23±0.39	0.2295
Global-Local	0.9980	2.39±0.02	0.0923	0.9884	5.54±0.18	0.1317
Our (Reg.)	**0.9988**	**2.61±0.04**	**0.0611**	-	-	-
Ours (Fit.)	0.9979	2.38±0.04	0.0764	**0.9888**	**6.26±0.27**	**0.1169**

Table 1. Comparison with the state-of-the-art methods on gesture-to-gesture translation, *i.e.,* GestureGAN [34], and pose-guided person generation, *i.e.,* Pose-Attn [39] and Global-Local [26]. Notably, due to the lack of 3D hand annotation in Senz3D, Ours (Reg.) cannot be applied on this dataset. ↑ indicates the higher the better, while ↓ indicates the lower the better.

Layers				Metrics		
D	*R1*	*R2*	*U*	SSIM ↑	IS ↑	LPIPS ↓
				0.9979	2.48±0.03	0.0759
✓	✓			**0.9982**	**2.55±0.03**	**0.0693**
	✓	✓		0.9980	2.39±0.03	0.0723
		✓	✓	0.9978	2.38±0.03	0.0769
✓	✓	✓	✓	0.9980	2.54±0.03	0.0733

Table 2. Ablation study on different locations inserting the SPADE layer on the STB Dataset. *D*, *R1*, *R2*, *U* represent the downsampling layers, the first half of the residual blocks, the latter half of the residual blocks and the upsampling layers, respectively. ↑ indicates the higher the better, while ↓ indicates the lower the better.

age x and the ground-truth image y, the SSIM metric is formulated as follows,

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (20)$$

where μ_x and μ_y are the average of image x and y while σ_x^2 and σ_y^2 are the variance of image x and y. σ_{xy} refers to the covariance of image x and y. c_1 and c_2 refer to two variables to stabilize the division with weak denominator. Given the generated image x, the IS metric is calculated as,

$$\mathrm{IS}(x) = e^{\mathbb{E}_x[D_{KL}(p(y|x)||p(y))]}, \quad (21)$$

where y is the label predicted by a pre-trained Inception model [33]. $p(y)$ and $p(y|x)$ refer to distribution of the label y and conditional distribution of y based on x, respectively. $D_{KL}(\cdot)$ denotes the K-L diversity. The LPIPS metric is defined as a weighted perceptual similarity between the generated image x and the ground-truth image y, which can be formulated as follows,

$$\mathrm{LPIPS}(x,y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (f_{hw}^{x,l} - f_{hw}^{y,l})\|_2^2, \quad (22)$$

where $f_{hw}^{x,l}$ and $f_{hw}^{y,l}$ are the feature of the generated image x and the ground-truth image y from layer l extracted by a pretrained AlexNet [17].

4.2. Comparison with the State-of-the-art Methods

We compare our method with several state-of-the-art methods on gesture-to-gesture translation [34] and pose-guided person generation [39, 26]. The quantitative results on STB and Senz3D are reported in Table 1. It demonstrates that our method outperforms previous methods. When compared with challenging Global-Loal [26], our method achieves a relative 33.8% and 11.2% gain under the LPIPS metric on STB and Senz3D, respectively.

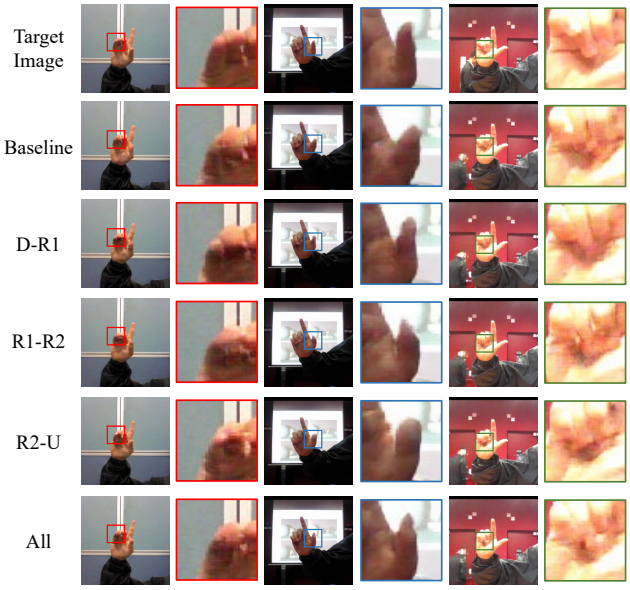

Figure 5. Qualitative comparison among different locations inserting the SPADE structure. *D*, *R1*, *R2* and *U* denote the downsampling layers, the first half of the residual blocks, the latter half of the residual blocks and the upsampling layers, respectively. Under the setting *D-R1*, the generated images demonstrate more fine-grained details and reasonable shading.

Furthermore, we make qualitative comparison with previous methods mentioned above. From Fig. 3, it can be observed that, due to the insufficient representation capability of 2D joints, previous methods cannot generate the images that exactly match the posture of the target images with fine-grained details. In contrast, by taking full advantage of expressive model-aware representation, the images generated by our method exhibit sharper edges and more details such as fingernail and palm print. This is because the transformation flow is more meticulous and precise compared to the flow learned by the network in previous methods.

In addition, as shown in Fig. 4, we visualize the generated video by translating the same source image to a target pose sequence in a frame-by-frame manner. Previous methods often suffer flickering and discontinuities, which is partially attributed to the semantically inaccurate 2D joints. In

contrast, our framework can generate more fluent video.

4.3. Ablation Study

In this subsection, we perform several ablation studies to verify the impact of different designs on the SPADE and attention sampler structure in our framework. We report the results in Table 2 and Table 3.

In the ablation of the SPADE structure, we take the network without SPADE as the basic setting, where the condition information is fed into the network by concatenating with the input image. As shown in Table 2, it can be observed that our method achieves the best performance when we adopt the SPADE structure on the encoding part, including the downsampling layers and the first half of the residual blocks. This is due to the fact that the condition fed into the network by concatenating with the input image is "washed away" during forward propagating of the network. The SPADE plays the role of constantly reminding the network aware of the structure information. However, due to the potential gap between the structure information and visual images, adopting the SPADE on the decoding part may have a misleading effect. Additionally, we analyze the visual results under these different settings. In Fig. 5, it can be seen that under the setting *D-R1*, which adopts the SPADE structure on the encoding part, the generated images contain the most fine-grained details and reasonable shading.

Based on the best setting in the ablation of SPADE structure, we further conduct the ablation study on the kernel size of the attention sampler. When the kernel size in the attention sampler is set to 1, the attention sampler deteriorates into the grid sampler. We take the network with kernel size 1 as our basic setting. From Table 3, it can be observed that networks with the attention sampler all outperform the network with the simple grid sampler. Among them, the performance reaches the top when the kernel size is equal to 5. When the kernel size is larger than 5, too large patch may contain some irrelevant disturbance resulting in inferior performance, and increase the computation cost notably. Furthermore, we compare the generated images under these different settings. In Fig. 6, it demonstrates that the generated images under the setting $k = 5$ have the most distinct silhouettes as well as vivid details, which is in line with the quantitative results. Unless stated, the kernel size is set to 5 in all experiments.

The above ablation study is performed on utilizing the directing regressing method to extract hand representation. We further compare different extraction methods in Table 1. We denote the direct regressing and iterative fitting method as Reg. and Fit., respectively. These two methods have their pros and cons. Fitting only relys on the frame-level 2D keypoints supervision and takes a relatively long processing time. Direct regressing has faster inference speed, however given the domain gap between datasets, the framework needs to be fine-tuned on the dataset with 3D annotations. In Table 1, it can be observed that these two methods achieve comparable performance on the STB dataset.

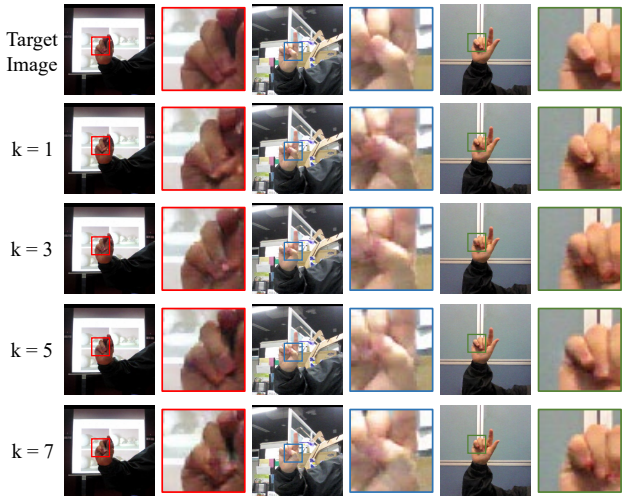

Figure 6. Qualitative comparison among different kernel sizes k of the attention sampler. Under the setting $k = 5$, the generated images have the sharpest edges and the best fine-grained details.

Kernel Size (k)	Metrics		
	SSIM ↑	IS ↑	LPIPS ↓
1	0.9982	2.55±0.03	0.0693
3	**0.9988**	2.40±0.03	0.0650
5	**0.9988**	**2.61±0.02**	**0.0611**
7	**0.9988**	2.53±0.04	0.0639

Table 3. Ablation study on the kernel size k of the attention sampler on STB. k refers to the size of extracted patch. ↑ indicates the higher the better, while ↓ indicates the lower the better.

5. Conclusion

In this paper, we introduce hand prior and propose the first model-aware framework to handle the task of gesture-to-gesture translation. Our framework first generates expressive model-aware hand representation from the gesture image and then builds the dense hand topology map used for calculating the transformation flow between the source and target. During gesture synthesis, we further emphasize the fine-grained structure by modulating the topology information in a spatially-adaptive way, jointly with the attention mechanism enhancing the final generated gesture image. Extensive experiments on two widely-used benchmarks demonstrate the superiority of our framework. Our method achieves state-of-the-art performance under SSIM, IS and LPIPS metrics, and shows better fine-grained structure than previous methods.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Contract U20A20183, 61632019 and 61836011, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497 . It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3D hand shape and pose from images in the wild. In *CVPR*, pages 10843–10852, 2019.

[2] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *ECCV*, pages 666–682, 2018.

[3] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. SO-HandNet: Self-organizing network for 3D hand pose estimation with semi-supervised learning. In *ICCV*, pages 6961–6970, 2019.

[4] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-based 3D hand pose estimation from monocular video. *IEEE TPAMI*, 33(9):1793–1805, 2011.

[5] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *CVPR*, pages 5154–5163, 2020.

[6] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational U-Net for conditional appearance and shape generation. In *CVPR*, pages 8857–8866, 2018.

[7] Stuart Geman and Donald E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the ISI*, 52(4):5–21, 1987.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[9] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. DeepCap: Monocular human performance capture using weak supervision. In *CVPR*, pages 5052–5063, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, pages 118–134, 2018.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[15] Ladislav Kavan and Jiří Žára. Spherical blend skinning: A real-time deformation of articulated models. In *ACM I3D*, pages 9–16, 2005.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, pages 1–14, 2013.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[18] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *CVPR*, pages 5904–5913, 2019.

[19] Yahui Liu, Marco De Nadai, Gloria Zen, Nicu Sebe, and Bruno Lepri. Gesture-to-gesture translation in the wild via category-independent conditional maps. In *ACM MM*, pages 1916–1924, 2019.

[20] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017.

[21] Alvise Memo and Pietro Zanuttigh. Head-mounted gesture controlled interface for human-computer interaction. *MTA*, 77(1):27–53, 2018.

[22] Tony Mullen. *Mastering blender*. John Wiley & Sons, 2011.

[23] Iason Oikonomidis, Manolis IA Lourakis, and Antonis A Argyros. Evolutionary quasi-random search for hand articulations tracking. In *CVPR*, pages 3422–3429, 2014.

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.

[25] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019.

[26] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *CVPR*, pages 7690–7699, 2020.

[27] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 36(6):245, 2017.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[29] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuiliere, and Nicu Sebe. Deformable GANs for pose-based human image generation. In *CVPR*, pages 3408–3416, 2018.

[30] Tomas Simon, Hanbyul Joo, Iain Matthews, and *et al.* Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, pages 1145–1153, 2017.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, pages 1–14, 2014.

[32] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*, pages 2456–2463, 2013.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[34] Hao Tang, Wei Wang, Dan Xu, Yan Yan, and Nicu Sebe. GestureGAN for hand gesture-to-gesture translation in the wild. In *ACM MM*, pages 774–782, 2018.

[35] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM TOG*, 35(6):1–11, 2016.

[36] Zhenyu Wu, Duc Hoang, Shih-Yao Lin, Yusheng Xie, Liangjian Chen, Yen-Yu Lin, Zhangyang Wang, and Wei Fan. MM-hand: 3D-aware multi-modal guided hand generation for 3D hand pose synthesis. In *ACM MM*, pages 2508–2516, 2020.

[37] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, and *et al.* A hand pose tracking benchmark from stereo matching. In *ICIP*, pages 982–986, 2017.

[38] Richard Zhang, Phillip Isola, and *et al.* The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[39] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, pages 2347–2356, 2019.

[40] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, pages 4903–4911, 2017.