Human from Blur: Human Pose Tracking from Blurry Images

Yiming Zhao\textsuperscript{1}  Denys Rozumnyi\textsuperscript{1}  Jie Song\textsuperscript{1}
Otmar Hilliges\textsuperscript{1} Marc Pollefeys\textsuperscript{1,2} Martin R. Oswald\textsuperscript{1,3}
\textsuperscript{1}ETH Zürich  \textsuperscript{2}Microsoft  \textsuperscript{3}University of Amsterdam

Abstract

We propose a method to estimate 3D human poses from substantially blurred images. The key idea is to tackle the inverse problem of image deblurring by modeling the forward problem with a 3D human model, a texture map, and a sequence of poses to describe human motion. The blurring process is then modeled by a temporal image aggregation step. Using a differentiable renderer, we can solve the inverse problem by backpropagating the pixel-wise re-projection error to recover the best human motion representation that explains a single or multiple input images. Since the image reconstruction loss alone is insufficient, we present additional regularization terms. To the best of our knowledge, we present the first method to tackle this problem. Our method consistently outperforms other methods on significantly blurry inputs since they lack one or multiple key functionalities that our method unifies, i.e. image deblurring with sub-frame accuracy and explicit 3D modeling of non-rigid human motion.

1. Introduction

Accurate tracking of human motion is often crucial for understanding dynamic scenes from images. Human motion estimation has a wide field of applications such as improving human-robot collaboration [2], human-machine interaction in general [12], better safety for autonomous driving [20], markerless human motion capture [36, 35, 27], sports analysis, and the movie and entertainment industry. A particular difficulty occurs when the human motion is fast, or low light conditions demand longer camera exposure times, which can both lead to blurry images from which it is significantly harder to estimate the human pose.

The main goal of our method is accurate 3D human pose tracking from substantially blurred images or videos. Hence, it is related to both human pose estimation and image deblurring methods. On the one hand, while there is a variety of methods that address 3D human pose estimation from RGB or RGB-D images, there is no method that is designed to handle substantially blurred images. Moreover, none of the human pose estimation methods is able to estimate human pose at sub-frame accuracy. On the other hand, there is a large amount of methods that aim at deblurring images and videos, but they mostly only assume simplified scenarios, e.g. without out-of-image-plane object rotations, or only for rigidly moving objects [45, 46]. So far, human pose estimation and image deblurring has not been studied jointly. Also, there is no public dataset to evaluate such task since none of standard datasets for human pose estimation include significant amounts of motion blur.

We propose the first method that recovers human pose at sub-frame accuracy from blurry inputs, even from a single blurry image (Fig. 1). We make the following contributions:

1. We present the first method for human pose estimation from substantially blurred images that recovers sub-frame accurate poses as well as texture and body shape.
2. We generate a synthetic dataset and collected real-world motion-blurred data of humans for evaluation purposes. We further propose corresponding evaluation metrics to assess and compare to future methods.
3. The proposed method only relies on test-time optimization and is learning-free, apart from the initialization.

Figure 1. Human from Blur (HfB) on a real-world sequence. Given a blurry image with human motion and the corresponding background, HfB recovers the human shape and sub-frame motion. We visualize sub-frame human pose and show the reconstructed mesh from a novel view.
and the motion prior, which is only needed for the single-frame case. Hence, our method does not require large amounts of annotated training data.

2. Related work

The proposed method is at the intersection of human pose tracking and image/video deblurring.

**3D Human pose estimation.** The 3D pose of a human is usually represented as a skeleton of 3D joints [34, 36, 35, 65, 51]. In order to obtain more fine-grained representations of the human body, parametric body models such as SCAPE [3] or the SMPL family [41, 42, 38] have been introduced to capture the 3D body pose. Iterative optimization-based approaches have been leveraged for model-based human pose estimation. [9, 48, 11, 4, 42] proposed to estimate the parameters of the human model by leveraging silhouettes or 2D keypoints. On the other hand, direct parameter regression via neural networks has been explored [15, 53, 56, 37, 10, 54, 58, 61, 24, 52, 27, 18]. Given a single RGB image, a deep network is used to regress the human model parameters. There is another line of work that combines the advantages of both optimization and regression to fit the SMPL body [19, 49]. Although there have been significant advances of human pose estimation from monocular images or videos, a method which is able to deal with blurry input is still missing.

**Image and video deblurring.** A large amount of methods have studied generic image [21, 22] and video deblurring, e.g. [14, 39, 62, 47, 64, 7, 23, 60, 50, 16, 43, 44]. Some attempts to specialize on deblurring depicted humans have already been made. For instance, [8] focuses only on deblurring human faces. Closely related to our problem setting, [31] addresses deblurring of human motion using an adversarial approach, which focuses on image deblurring rather than pose estimation, and it does not recover at sub-frame accuracy. The follow-up method [30] generalizes to joint human motion and scene deblurring with a similar methodology, but the sub-frame poses are never recovered.

The proposed method is partially inspired by Shape from Blur (SfB) [45], which uses a similar test-time optimization to recover 3D shape and sub-frame motion of simple rigid objects with spherical topology from a single blurry image with a given background. Motion from Blur (MfB) [46] extends SfB to multiple video frames. There is also a related Animation from Blur method [63], but it assumes a motion guidance is provided as an additional input.

3. Method

The inputs to our method are an image $I$ with the blurred human and the corresponding clean background image $B$. The desired output is a human shape parameter $\beta$, texture image $T$, and three functions representing sub-frame human motion that depend on a timestamp $i$. This timestamp represents the sub-frame time interval and is defined between $1$ and $N$, where $N$ is the desired number of sub-frames. Effectively, it means that we generate a temporal super-resolution or a short video with $N$ frames out of each single input frame. Those three functions are human body translation $T_i$, rotation $R_i$, and sub-frame human pose $\theta_i$ that represents joint rotation. They are all represented by a set of low-degree polynomials, where translations and rotations have each four degrees of freedom (direction with
distance and axis with angle). This polynomial representation generates poses in a strict chronological order and is continuous, differentiable, and can be easily initialized with a given initial pose (Sec. 3.3). Human pose and shape representations follow the SMPL human model [41]. The texture image $\mathcal{T}$ is mapped using a fixed UV mapping from SMPL.

As the first step, we generate the human SMPL mesh $\Theta_i$ at timestamp $i$ with a given pose, shape, and texture parameters (Fig. 2). Then, we move the whole mesh $\Theta_i$ by translation $T_i$ and rotation $R_i$ given by motion function $\mathcal{M}$:

$$\Theta_i = \mathcal{M}(\text{SMPL}(\theta_i, \beta, \mathcal{T}), T_i, R_i).$$ (1)

To render the sub-frame silhouette and appearance of the mesh, we use Differentiable Interpolation-based Renderer (DIB-R) [6]. This differentiable rendering provides two outputs. The first one is appearance rendering $\mathcal{R}_F(\Theta_i)$ that outputs projected human appearance. The second one is silhouette rendering $\mathcal{R}_S(\Theta_i)$ that outputs projected human silhouette. In this work, we assume a static camera.

**Image formation model.** Given all previously defined parameters, we can finally define the image formation model. It follows a standard alpha matting approach:

$$\hat{I} = \left(1 - \frac{1}{N} \sum_{i=1}^{N} \mathcal{R}_S(\Theta_i)\right) \cdot B + \frac{1}{N} \sum_{i=1}^{N} \mathcal{R}_S(\Theta_i) \cdot \mathcal{R}_F(\Theta_i).$$ (2)

The generated image $\hat{I}$ consists of the background image, scaled down by the inverse alpha channel, and the blurred foreground human body. The alpha channel is modeled by averaging all projected sub-frame silhouettes.

### 3.1. Loss terms

The key components of our method are the image formation loss and the matting loss. The image formation loss forces the reconstructed image to be as close as possible to the input image. The matting loss favors silhouettes that are consistent with the initially estimated alpha channel. The other losses are auxiliary and regularization terms that make the optimization easier and refine the final results.

**Image formation loss.** This loss measures the input image reconstruction according to the image formation model (2). We compute the mean squared error between the observed input image and our reconstruction as:

$$\mathcal{L}_I = |I - \hat{I}|_2.$$ (3)

**Matting loss.** If the image formation loss (3) is the sole loss to be minimized, the optimization becomes extremely difficult. Experimentally, such optimization is ambiguous and mostly results in an undesired local minimum. Therefore, we further impose a loss on our approximated rendered alpha channel, which is computed as the average of sub-frame silhouettes, $\alpha_{\text{target}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{R}_S(\Theta_i)$, according to the image formation model (2). The initial alpha channel $\alpha_{\text{in}}$ is estimated using a pre-trained Background-Matting-V2 [28] model, based on the input blurry image and the corresponding background. Finally, the matting loss computes the intersection over union between our rendered alpha channel from averaging and the one from [28]:

$$\mathcal{L}_\alpha = 1 - \frac{\min(\alpha_{\text{in}}, \alpha_{\text{target}})}{\max(\alpha_{\text{in}}, \alpha_{\text{target}})}.$$ (4)

where the intersection over union for non-binary inputs is a ratio between the sum of pixel-wise min and max operators.

**Surface texture smoothness.** The UV texture map from SMPL contains many non-overlapping regions (see Fig. 3, HIB row), and the correct neighborhoods are not properly defined. Therefore, the commonly used total variation loss for texture smoothness [46] cannot be directly applied in this case since it will propagate the color of the void area. To address this issue, we propose a surface texture smoothness term that accounts for the mesh faces neighborhood. For a given texture pixel $p_k$ and its 8 surrounding neighboring pixels $p_j \in \mathcal{N}(p_k)$, we pick those ones that are neighbors in the mesh ($c_{k,j} = 1$), i.e. they belong to adjacent triangular faces, and that are visible in at least one of the sub-frames ($\mathcal{V}_j = 1$). Then, we compute the cosine value between the face normal $n_k$ of the current pixel and the face normal $n_j$ of its chosen neighbors. The introduction of the cosine of face normals takes into account the mesh geometry, i.e. the texture should be smoother on flat surfaces. Then, the surface texture smoothness is expressed as a weighted sum of absolute differences in RGB pixels:

$$\mathcal{L}_S = \frac{1}{8|\mathcal{T}|} \sum_{p_k \in \mathcal{T}} \sum_{p_j \in \mathcal{N}(p_k)} c_{k,j} \mathcal{V}_j \cos(\langle n_k, n_j \rangle) |p_k - p_j|_1.$$ (5)

**Pose prior loss.** We import the pose prior loss from SMPLify-X [42]. This prior scores how feasible are the estimated pose parameters $\theta_i$:

$$\mathcal{L}_P = \frac{1}{N} \sum_{i=1}^{N} \text{prior}(\theta_i).$$ (6)

**SMPL shape regularization.** We add norm regularization on the SMPL shape parameter $\beta$ to avoid irregular human body shape as used in SMPL [41]:

$$\mathcal{L}_\beta = |\beta|^2_2.$$ (7)

**Polynomial regularization.** The polynomial coefficients of the pose, translation, and rotation could be serialized into a matrix $C \in \mathbb{R}^{d}$, where $d$ is the degree of the polynomial, and $J$ is the number of joints in the SMPL model.
The whole body translation and rotation are already incorporated into matrix $C$, thus we have $4(J+2)$ polynomials of $d$ degree. Since rotations, translations, and joint poses have 4 degrees of freedom each, we have a separate polynomial for each degree of freedom. We apply both the L1-norm and the Frobenius norm on the polynomial coefficients:

$$\mathcal{L}_C = |C|_1 + |C|_F .$$

The intention of adding this regularization is to avoid extreme joint movement.

**Background regularization.** We assume that the human texture is sufficiently distinct from the background. This is enforced by the difference between the projected object appearance and the background:

$$\mathcal{L}_B = \frac{1}{N} \sum_{i=1}^N \frac{1}{|B - \mathcal{R}_F(\Theta_i)| + \epsilon} \text{ with } \epsilon = 10^{-6} .$$

**Adversarial short motion prior.** Since the human body consists of multiple joints, there exists a significant amount of ambiguity in case of a single input blurry image. The ambiguity comes mainly from the unknown motion direction. In fact, both the forward and the backward directions provide the same blurry image according to the image formation model (2). Potentially, there are exponentially many motion directions for each joint that lead to the same input. And it is infeasible to estimate the correct direction directly from a single image without any additional priors. Otherwise, the choice of motion direction will be arbitrary. Many prior studies [42, 18] offer motion priors, but they are not suitable for our setting.

To address this problem we propose the adversarial motion prior to recognize wrong (reversed) motion of joints. Based on our polynomial motion representation, we propose an adversarial model that could supervise on the polynomial coefficients $C$. The model is inspired by the image in-painting methods [40, 29].

Our adversarial model consists of two components: a discriminator $D$ that generates a binary indicator function to identify unrealistic entries in the coefficients $C$, and a correction-generator $G$ that predicts realistic polynomial coefficients from the given polynomial coefficients $C$ and the indicator function $I_c$.

The training data are sampled from the AMASS dataset [33] (CMU [5] and ACCAD [1]). The training is supervised jointly by four loss terms. The discriminator loss is the binary cross entropy loss, which is applied to the indicator function predicted by the discriminator and compared to the ground truth. The generator loss, specifically...
the reconstruction loss, comprises three terms. The first one is L1 loss between the coefficient matrix predicted by the correction- generator and the ground truth. The second is L2 loss between the reconstructed pose and the ground truth pose. The last one is the mean per joint position error (MPJPE) [49] between the reconstructed joint positions and the ground truth SMPL joint positions.

This adversarial model is pre-trained as mentioned above and is fixed during optimization. In case of a single input blurry image, the adversarial motion prior is incorporated into the optimization as the L1 loss between the generator output and the polynomial coefficients $C$:

$$L_A = |G(D(C)), C) - C|_1.$$  \hspace{1cm} (10)

**Joint loss.** The final loss is a weighted sum of all previously defined losses:

$$L = w_I L_I + w_a L_a + w_S L_S + w_p L_P + w_\beta L_\beta + w_C L_C + w_B L_B + w_A L_A.$$  \hspace{1cm} (11)

### 3.2. Multiple blurry images

Our approach can be extended to multiple consecutive blurry images in a video. In this case, the human body shape $\beta$ and texture $T$ are assumed to be the same for all input images, while the other parameters, e.g., poses, are separate for each frame. In general, this setting is simpler since there are more ambiguity in the motion direction of each joint. Therefore, the adversarial motion prior $L_A$ is not needed anymore.

For smooth joint motion in consecutive frames, we add a boundary restriction on the joints rotation and position. For instance, in case of two input blurry images, we add a boundary restriction at the end timestamp $i = N$ of the first image and the start timestamp $i = 1$ of the second image. The boundary restriction forces the joints rotation and position and their first order derivatives to be equal at the boundary to preserve the motion continuity, and it is implemented by the L1 loss with a unit weight. For images with exposure gap, we extend the end timestamp of first image with exposure time $\tau$ (measured in sub-frames) and then apply boundary restriction at $N + \tau$.

### 3.3. Optimization

The joint loss (11) is minimized using the ADAM optimizer [17] for 200 iterations with learning rate 0.01 on a single 12 GB RTX 2080 Ti graphics card.

**Initialization.** To initialize our method, we use the METRO [27] human pose estimation method, which reconstructs a single human pose from a blurry image reasonably well, albeit without sub-frame accuracy. We fit the initial body translation, rotation, pose, and shape parameters to

![Figure 4. Comparison on synthetic data. Given a blurry image with human motion, sub-frame human poses generated by HfB are consistent, whereas for learning-based temporal super-resolution methods [13, 63] (with METRO [27] human poses on sub-frames) the poses are not consistent, e.g. motion of the right arm. We also visualize the raw METRO [27] pose prediction on the input blurry image.](image)
than one. We use this blur rate to classify the experiments.

When the human moves over a distance larger than its size within one blurry frame, the blur rate value is zero. When the human stays still, the blur rate value is one. We use this blur rate to classify the experiments.

In the end, we average these rendered silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, the blur rate value is one. We use this blur rate to classify the experiments.

Among the chosen baselines, we selected SfB [45] and MfB [46], designed for simple objects sub-frame deblurring and 3D reconstruction. Then, we compare our method to general temporal super-resolution methods, i.e. Jin et al. [15] and Animation-from-Blur (AIB) [63] for single frame experiments and Blurry Video Frame Interpolation (BIN) [47] for multi-frame experiments. To make them competitive, we also apply 3D human pose estimation METRO [27] on top of their deblurred sub-frames (temporal super-resolution), except for SfB and MfB, where the output sub-frames are of low quality, and human pose estimation methods do not detect anything.

**Blur rate.** In general, motion blur is determined by many factors. However, the main factors are the camera exposure time and the speed of the object motion. Even with those two factors, it is still a challenging task to quantify the exact amount of motion blur. In order to measure the approximate blur level, we define blur rate as:

\[
\text{blur rate} = \frac{\sum_{i=1}^{N} R_S(\Theta_i)}{\sum_{i=1}^{N} |R_S(\Theta_i)|} - 1. \tag{12}
\]

Here, we compute the union of all projected sub-frame silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, i.e. there is no overlap between the rendered silhouettes at the first and last timestamps, the blur rate is larger than one. We use this blur rate to classify the experiments.

In the end, we average these rendered silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, the blur rate value is one. We use this blur rate to classify the experiments.

**4. Experiments**

Among the chosen baselines, we selected SfB [45] and MfB [46], designed for simple objects sub-frame deblurring and 3D reconstruction. Then, we compare our method to general temporal super-resolution methods, i.e. Jin et al. [15] and Animation-from-Blur (AIB) [63] for single frame experiments and Blurry Video Frame Interpolation (BIN) [47] for multi-frame experiments. To make them competitive, we also apply 3D human pose estimation METRO [27] on top of their deblurred sub-frames (temporal super-resolution), except for SfB and MfB, where the output sub-frames are of low quality, and human pose estimation methods do not detect anything.

**Blur rate.** In general, motion blur is determined by many factors. However, the main factors are the camera exposure time and the speed of the object motion. Even with those two factors, it is still a challenging task to quantify the exact amount of motion blur. In order to measure the approximate blur level, we define blur rate as:

\[
\text{blur rate} = \frac{\sum_{i=1}^{N} R_S(\Theta_i)}{\sum_{i=1}^{N} |R_S(\Theta_i)|} - 1. \tag{12}
\]

Here, we compute the union of all projected sub-frame silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, i.e. there is no overlap between the rendered silhouettes at the first and last timestamps, the blur rate is larger than one. We use this blur rate to classify the experiments.

In the end, we average these rendered silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, the blur rate value is one. We use this blur rate to classify the experiments.

Among the chosen baselines, we selected SfB [45] and MfB [46], designed for simple objects sub-frame deblurring and 3D reconstruction. Then, we compare our method to general temporal super-resolution methods, i.e. Jin et al. [15] and Animation-from-Blur (AIB) [63] for single frame experiments and Blurry Video Frame Interpolation (BIN) [47] for multi-frame experiments. To make them competitive, we also apply 3D human pose estimation METRO [27] on top of their deblurred sub-frames (temporal super-resolution), except for SfB and MfB, where the output sub-frames are of low quality, and human pose estimation methods do not detect anything.

**Blur rate.** In general, motion blur is determined by many factors. However, the main factors are the camera exposure time and the speed of the object motion. Even with those two factors, it is still a challenging task to quantify the exact amount of motion blur. In order to measure the approximate blur level, we define blur rate as:

\[
\text{blur rate} = \frac{\sum_{i=1}^{N} R_S(\Theta_i)}{\sum_{i=1}^{N} |R_S(\Theta_i)|} - 1. \tag{12}
\]

Here, we compute the union of all projected sub-frame silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, i.e. there is no overlap between the rendered silhouettes at the first and last timestamps, the blur rate is larger than one. We use this blur rate to classify the experiments.

Among the chosen baselines, we selected SfB [45] and MfB [46], designed for simple objects sub-frame deblurring and 3D reconstruction. Then, we compare our method to general temporal super-resolution methods, i.e. Jin et al. [15] and Animation-from-Blur (AIB) [63] for single frame experiments and Blurry Video Frame Interpolation (BIN) [47] for multi-frame experiments. To make them competitive, we also apply 3D human pose estimation METRO [27] on top of their deblurred sub-frames (temporal super-resolution), except for SfB and MfB, where the output sub-frames are of low quality, and human pose estimation methods do not detect anything.

**Blur rate.** In general, motion blur is determined by many factors. However, the main factors are the camera exposure time and the speed of the object motion. Even with those two factors, it is still a challenging task to quantify the exact amount of motion blur. In order to measure the approximate blur level, we define blur rate as:

\[
\text{blur rate} = \frac{\sum_{i=1}^{N} R_S(\Theta_i)}{\sum_{i=1}^{N} |R_S(\Theta_i)|} - 1. \tag{12}
\]

Here, we compute the union of all projected sub-frame silhouettes and divide it by the first silhouette. When the human stays still, the blur rate value is zero. When the human moves over a distance larger than its size within one blurry frame, i.e. there is no overlap between the rendered silhouettes at the first and last timestamps, the blur rate is larger than one. We use this blur rate to classify the experiments.
by up-sampling and interpolation at a high frame rate of 600 fps. In total, we generated 1861 blurry images for a single frame experiment and 305 short videos with two video frames to evaluate our multi-frame setting.

**Evaluation metrics.** We evaluate HfB on the joint position error in millimeters: Mean Per Joint Position Error (MPJPE) and Procrustes Analysis MPJPE (PA-MPJPE) as in [49]. For MPJPE, we initially align the coordinate axis orientation of the predicted motion sequence with the ground truth. For comparison to SfB and MfB, we also measure the intersection-over-union (IoU) between the generated silhouette and the ground truth one.

### 4.2. Results on BT-AMASS

First, we compare our method on the generated BT-AMASS dataset to the following baselines: SfB [45], MfB [46], and static interpolated METRO [27] (1124 single frames and 305 two-frames). The single-frame results with 1124 images are shown in Table 1, whereas multi-frame results in Table 2. Our method outperforms all baselines by a wide margin, especially for higher blur rate intervals. As expected, the performance steadily decreases with the increased blur rate. Additionally, we evaluate the influence of the Adversarial Motion Prior (AMP), which is only used for single-frame experiment. This prior improves results only for higher blur rates, whereas for lower blur rates, where there is less ambiguity, it is harmful.

### 4.3. Results on BC-CAPE

Next, we evaluate the proposed method on the generated BC-CAPE dataset, which contains 609 single frames and 205 short sequences with 4-frames. In this case, we compare to three temporal super-resolution methods: AfB [63], Jin et al. [13], and BIN [47]. For fair comparison, we augmented their sub-frame output with human pose estimation methods, either METRO [27] or HybrIK [24]. As shown in Tables 5 and 4, the proposed Human from Blur (HfB) method outperforms these baselines by a large margin, especially on larger blur rates. The performance gain is even higher for the multi-frame experiment (Table 5).

### 4.4. Real dataset B-AIST++

Finally, we evaluate on the real-world dataset with various human motion and garments: B-AIST++ [63]. They use frame interpolation to generate high-speed frames from original dancing dataset AIST++ [26]. B-AIST++ provides significantly blurred images with human motion. We generate ground-truth sub-frame human pose by running METRO [27] on top of the ground-truth sub-frames. Table 3 shows that our method outperforms other baselines when 3 consecutive frames are averaged, which translates to blur rate 0.45. Note that AfB [63] is trained on this dataset.
Figure 6. Multi-frame evaluation. We compared to BIN [47], with METRO [27] human poses on top of their sub-frames. The visual results show that BIN fails, however METRO is still robust to some amount of blur and detects human poses, which are not consistent over time. The proposed method generates motion which is more consistent with the ground truth.

<table>
<thead>
<tr>
<th>blur rate</th>
<th>&lt;0.2</th>
<th>[0.2,0.3]</th>
<th>[0.3,0.4]</th>
<th>[0.4,0.5]</th>
<th>[0.5,0.6]</th>
<th>[0.6,0.8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
</tr>
<tr>
<td>HfB (ours)</td>
<td>75.2</td>
<td>81.2</td>
<td>76.4</td>
<td>83.0</td>
<td>84.6</td>
<td>94.2</td>
</tr>
<tr>
<td>AfB [63] + METRO [27]</td>
<td>82.4</td>
<td>89.1</td>
<td>84.1</td>
<td>89.2</td>
<td>90.6</td>
<td>100.8</td>
</tr>
<tr>
<td>Jin et al. [14] + METRO [27]</td>
<td>74.4</td>
<td>77.2</td>
<td>76.2</td>
<td>82.0</td>
<td>82.6</td>
<td>90.8</td>
</tr>
<tr>
<td>Jin et al. [14] + PyMaF [59]</td>
<td>79.9</td>
<td>83.6</td>
<td>80.3</td>
<td>85.5</td>
<td>93.4</td>
<td>106.1</td>
</tr>
<tr>
<td>AfB [63] + PyMaF [59]</td>
<td>83.5</td>
<td>87.0</td>
<td>86.7</td>
<td>91.42</td>
<td>93.7</td>
<td>109.8</td>
</tr>
</tbody>
</table>

Table 4. Results on Blurred-Clothed CAPE dataset. Our method outperforms competitive baselines on larger blur rates.

<table>
<thead>
<tr>
<th>blur rate</th>
<th>&lt;0.2</th>
<th>[0.2,0.3]</th>
<th>[0.3,0.4]</th>
<th>[0.4,0.5]</th>
<th>[0.5,0.7]</th>
<th>[0.7,0.9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
<td>PA-MPJPE ↓</td>
<td>MPJPE ↓</td>
</tr>
<tr>
<td>HfB (ours)</td>
<td>83.7</td>
<td>85.3</td>
<td>86.1</td>
<td>91.3</td>
<td>90.5</td>
<td>96.4</td>
</tr>
<tr>
<td>BIN [47] + HybrIK [24]</td>
<td>76.6</td>
<td>78.8</td>
<td>86.7</td>
<td>96.5</td>
<td>93.5</td>
<td>106.9</td>
</tr>
<tr>
<td>AfB [63] + HybrIK [24]</td>
<td>82.4</td>
<td>84.6</td>
<td>96.8</td>
<td>105.3</td>
<td>100.8</td>
<td>111.1</td>
</tr>
<tr>
<td>BIN [47] + METRO [27]</td>
<td>84.0</td>
<td>85.9</td>
<td>88.5</td>
<td>97.5</td>
<td>94.0</td>
<td>106.8</td>
</tr>
<tr>
<td>AfB [63] + METRO [27]</td>
<td>87.7</td>
<td>90.2</td>
<td>90.8</td>
<td>98.1</td>
<td>97.7</td>
<td>108.9</td>
</tr>
<tr>
<td>PyMaF [59]</td>
<td>97.7</td>
<td>119.5</td>
<td>118.7</td>
<td>152.5</td>
<td>128.8</td>
<td>172.0</td>
</tr>
<tr>
<td>BIN [47] + PyMaF [59]</td>
<td>78.29</td>
<td>98.04</td>
<td>95.05</td>
<td>126.2</td>
<td>105.2</td>
<td>151.2</td>
</tr>
</tbody>
</table>

Table 5. Results on Blurred-Clothed CAPE dataset with 4 consecutive frames. We also combine both METRO [27] and HybrIK [24] with two baselines (BIN [47] and AfB [63]) to show the impact of different human pose estimation methods. We also show results of BIN [47] with PyMaF [59], which includes the interpolation of the joint positions by directly applying on blur frames. With multiple input frames, HfB outperforms other baselines for almost all blur rates (0.2 and higher).

whereas our method is purely optimization based.

4.5. Captured data

We captured 21 real-world sequences with significant amounts of motion blur, including four male and one female subject. The used cameras are the IDS camera and a GoPro 7, which were deliberately set at a low frame rate of 30 fps with exposure time of 30 ms to 50 ms. The recorded humans were asked to move fast. Background images are captured as well. As shown in Fig. 3 and Fig. 1, the final reconstructions are plausible. When compared to SfB [45], our method achieves significantly better results.
5. Conclusion

We proposed the first method to reconstruct sub-frame human motion and textured shape from substantially blurred images. The key idea is to approach the problem from a generative viewpoint and describe a fully differentiable forward process to generate blurry images from a given 3D human motion model. The core of our method is an image reconstruction loss that allows to solve the inverse problem with standard gradient descent methods. Experiments showed that the proposed method achieves the best results on both synthetic and real blurry data.

Acknowledgements. This research was supported by a Google Focused Research Award, Innosuisse grant No. 34475.1 IP-ICT, and a research grant by FIFA.

References

[1] Advanced Computing Center for the Arts and Design. AC-CAD MoCap Dataset. 4, 6
[5] Carnegie Mellon University. CMU MoCap Dataset. 4, 6
[25] Jizhizi Li, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. Bridging composite and real: Towards end-to-end deep


