

Photo Wake-Up: 3D Character Animation from a Single Photo

ECCV 2019

00.

INTRO

Photo Wake-up

<https://youtu.be/G63goXc5MyU>

00.

INTRO

Photo Wake-up



Input Photo



Photo Animation



Augmented Reality

00.

INTRO

Photo Wake-up

an application of viewing and animating humans in single photos in 3D

a novel 2D warping method to deform a posable template body model
to fit the person's complex silhouette to create an animatable mesh

a method for handling partial self occlusions

1. Mesh construction



2. Self occlusion

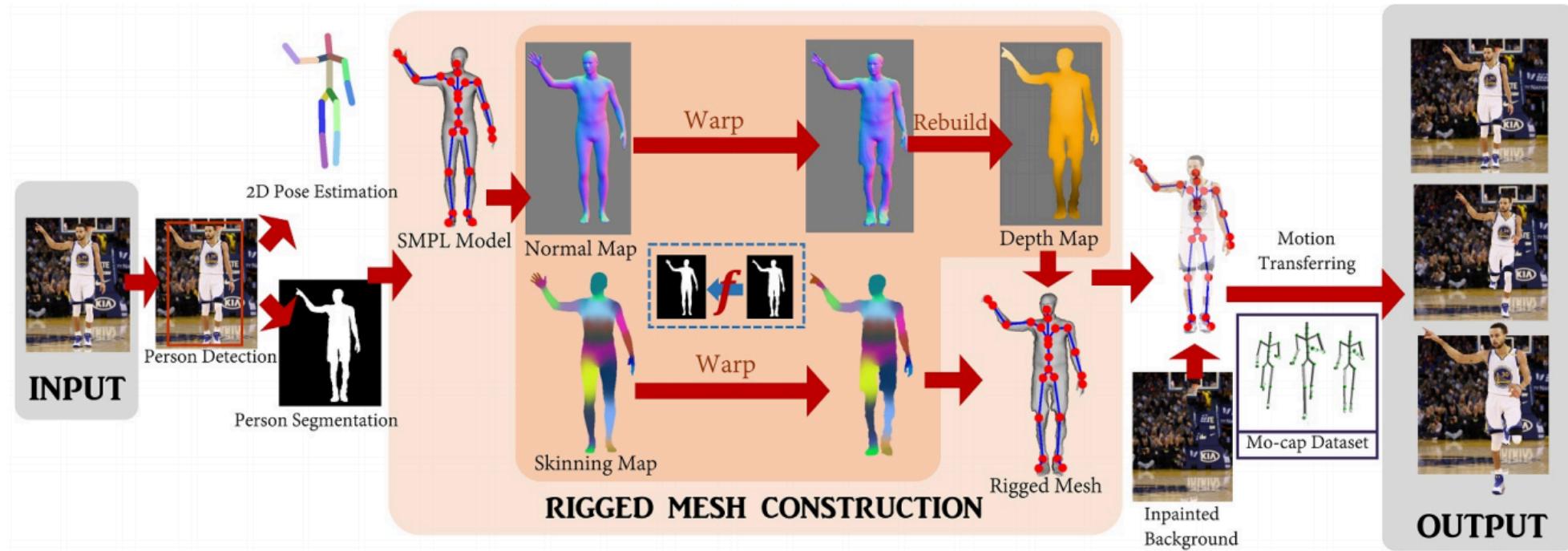


3. Final steps

1. Mesh construction & Rigging

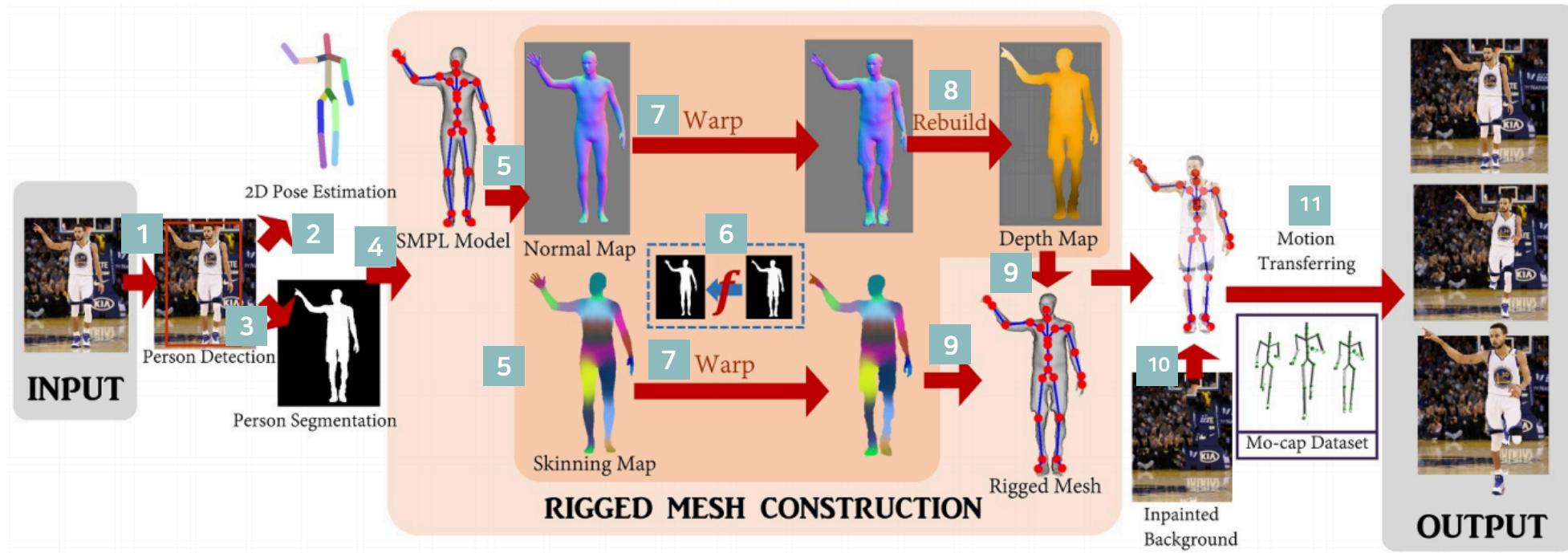
01. Mesh construction & Rigging

Introduction



01. Mesh construction & Rigging

Introduction



01. Mesh construction & Rigging

1) Person detection

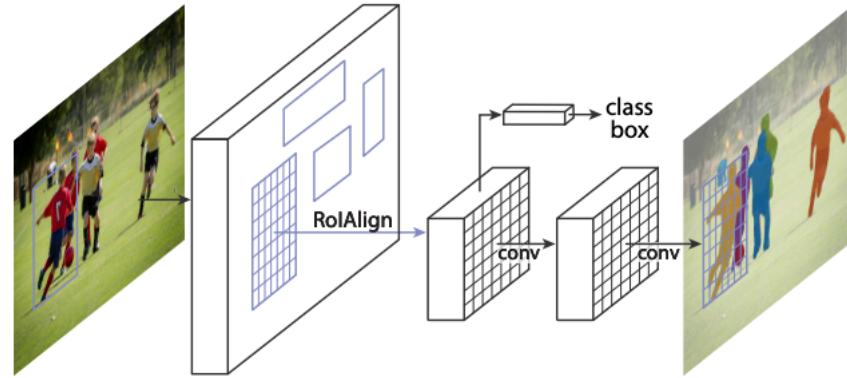
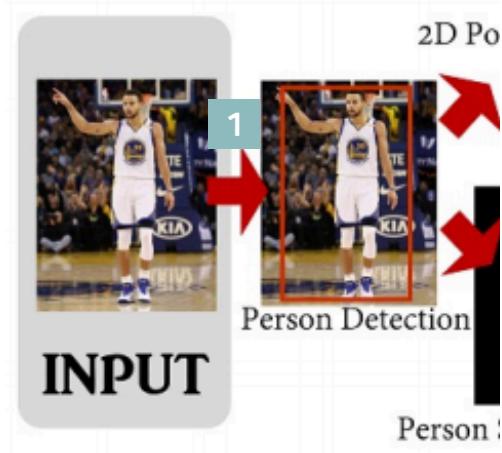


Figure 1. The **Mask R-CNN** framework for instance segmentation.



Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [15], achieving a *mask AP* of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

01. Mesh construction & Rigging

2) 2D Pose estimation

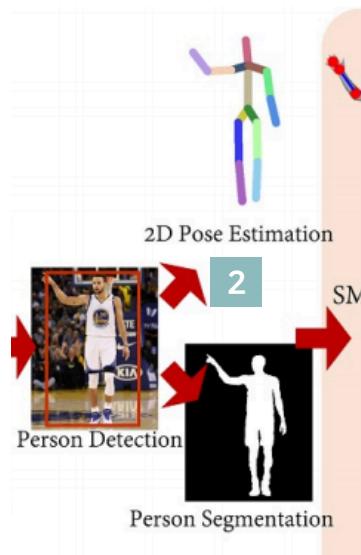


Figure 10: Qualitative results of our method on the MPII, LSP and FLIC datasets respectively. We see that the method is able to handle non-standard poses and resolve ambiguities between symmetric parts for a variety of different relative camera views.

01. Mesh construction & Rigging

3) Person Segmentation

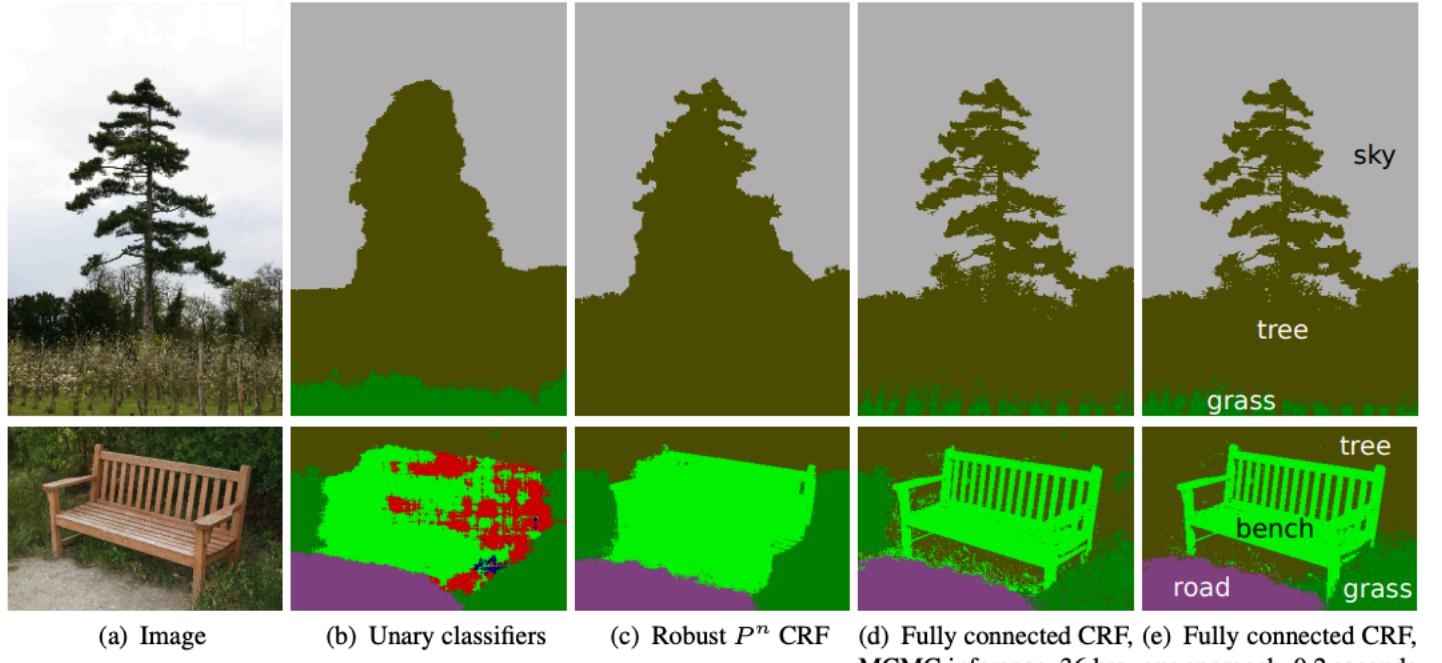
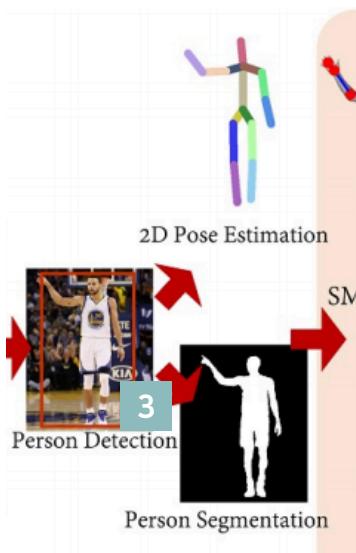


Figure 1: Pixel-level classification with a fully connected CRF. (a) Input image from the MSRC-21 dataset. (b) The response of unary classifiers used by our models. (c) Classification produced by the Robust P^n CRF [9]. (d) Classification produced by MCMC inference [17] in a fully connected pixel-level CRF model; the algorithm was run for 36 hours and only partially converged for the bottom image. (e) Classification produced by our inference algorithm in the fully connected model in 0.2 seconds.

01. Mesh construction & Rigging

4) SMPL model

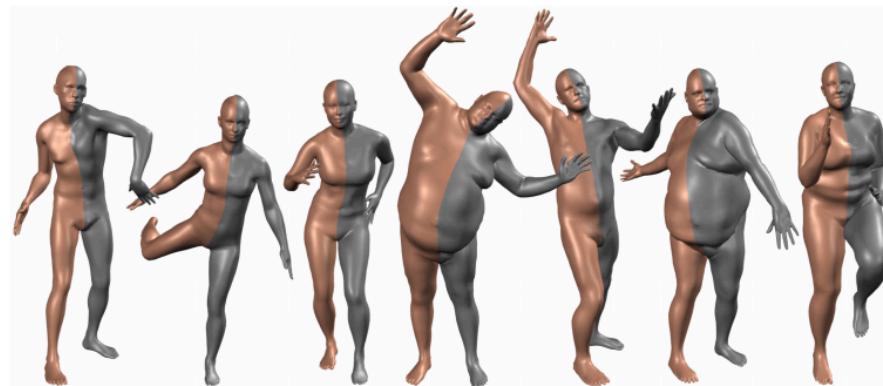
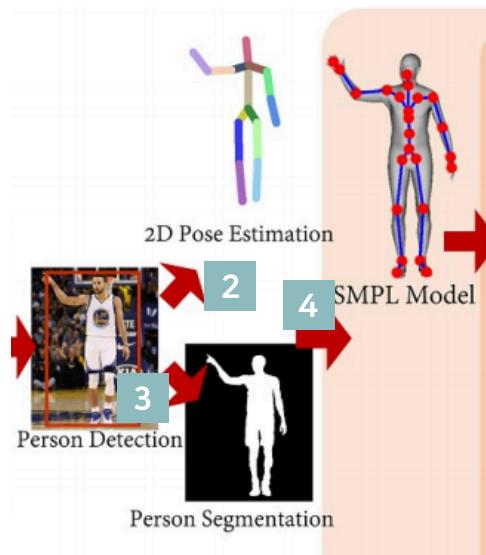
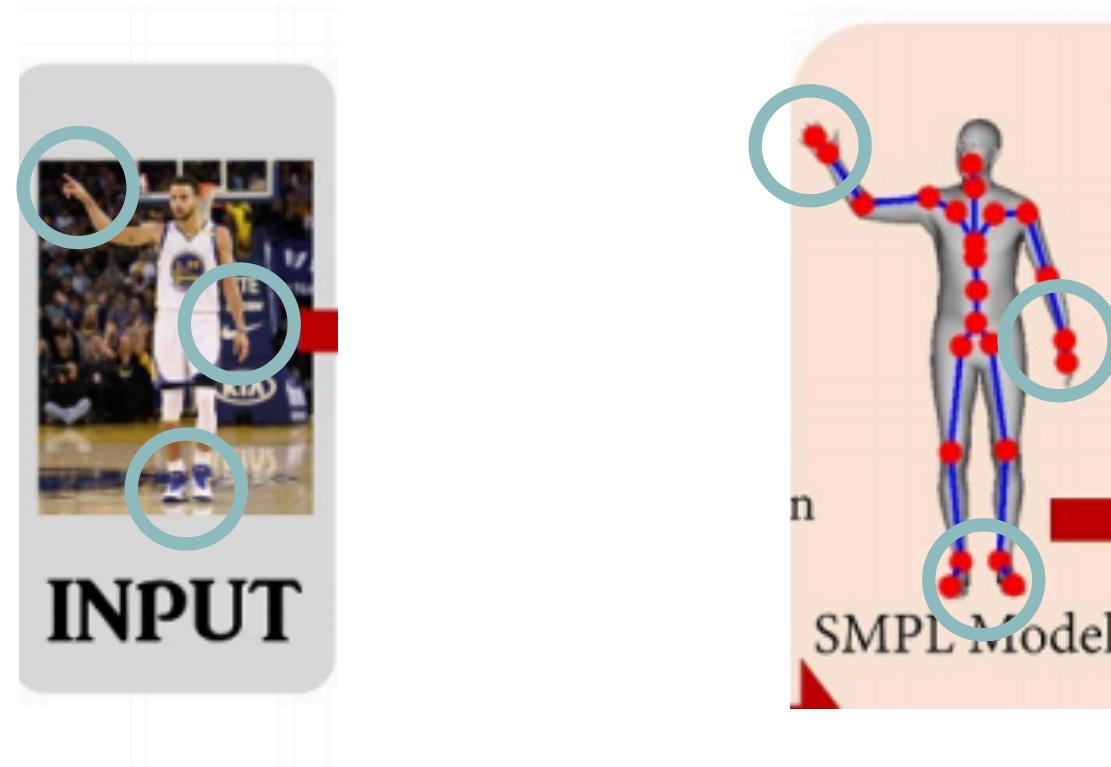


Figure 1: SMPL is a realistic learned model of human body shape and pose that is compatible with existing rendering engines, allows animator control, and is available for research purposes. (left) SMPL model (orange) fit to ground truth 3D meshes (gray). (right) Unity 5.0 game engine screenshot showing bodies from the CAESAR dataset animated in real time.

01. Mesh construction & Rigging

4) SMPL model - 모델의 한계



01. Mesh construction & Rigging

4) SMPL model - Solution

Bogo, Alldieck : SMPL모델을 만든 다음 그 모델이 실루엣에 맞도록 세부 조정



2D approach

- Match the person shilhouette in the original image
- Warp the projected SMPL normal & skinning map
- Construct front & back view and lift them into 3D along with 3D skeleton

01. Mesh construction & Rigging

4) SMPL model - solution

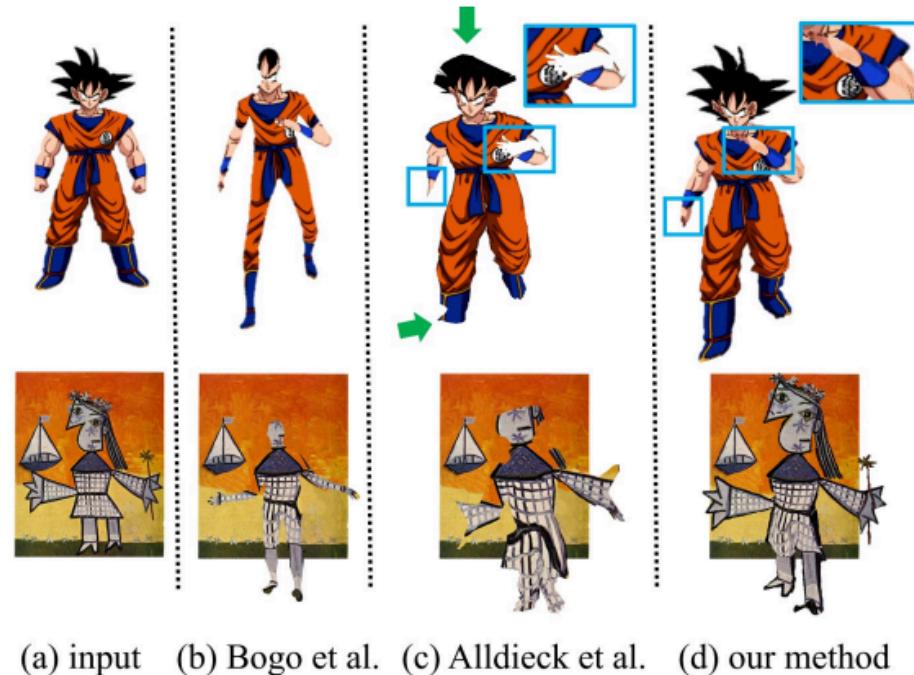
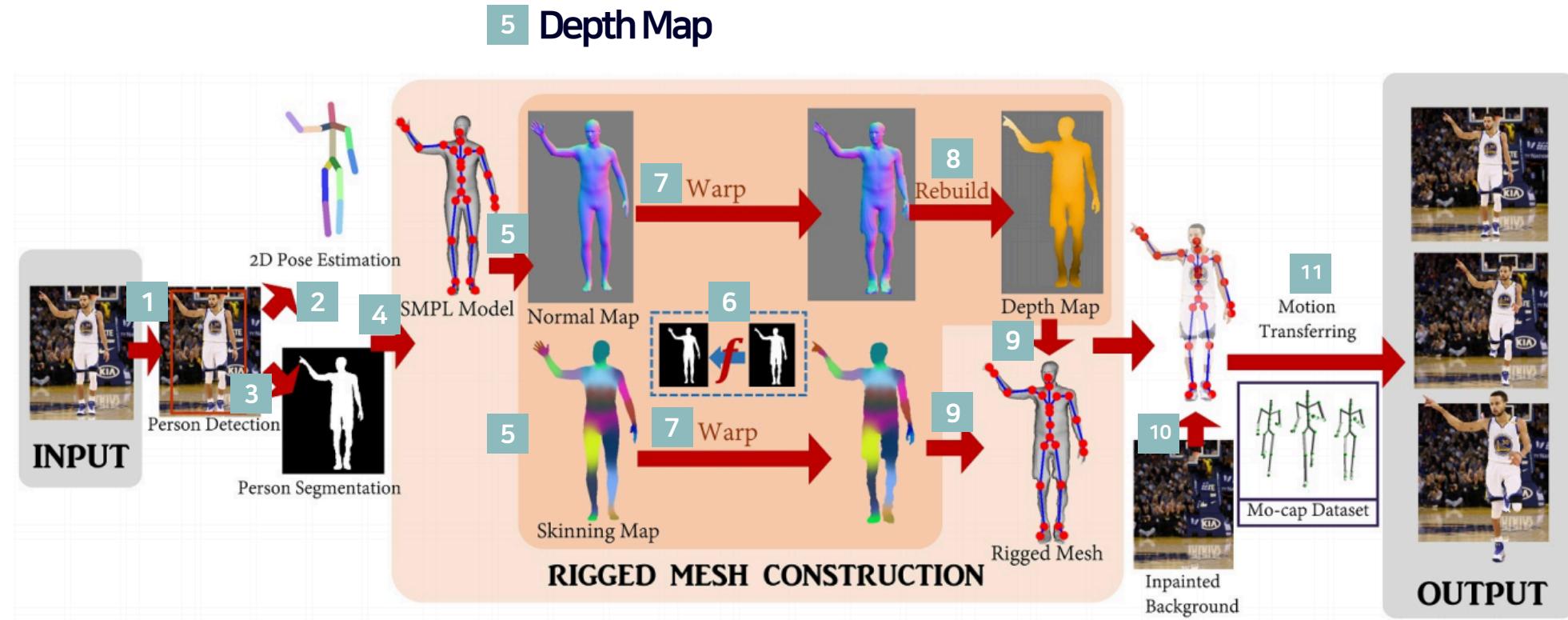


Figure 11: Comparison with [16, 11]. (a) Input photo. (b) A fitted SMPL model [16]. (c) A deformed mesh using [11]. The mesh fails to deform hair and shoes (green arrows) and fingers (blue box). (d) Our mesh. *Photo credits: [2, 8]*

01. Mesh construction & Rigging

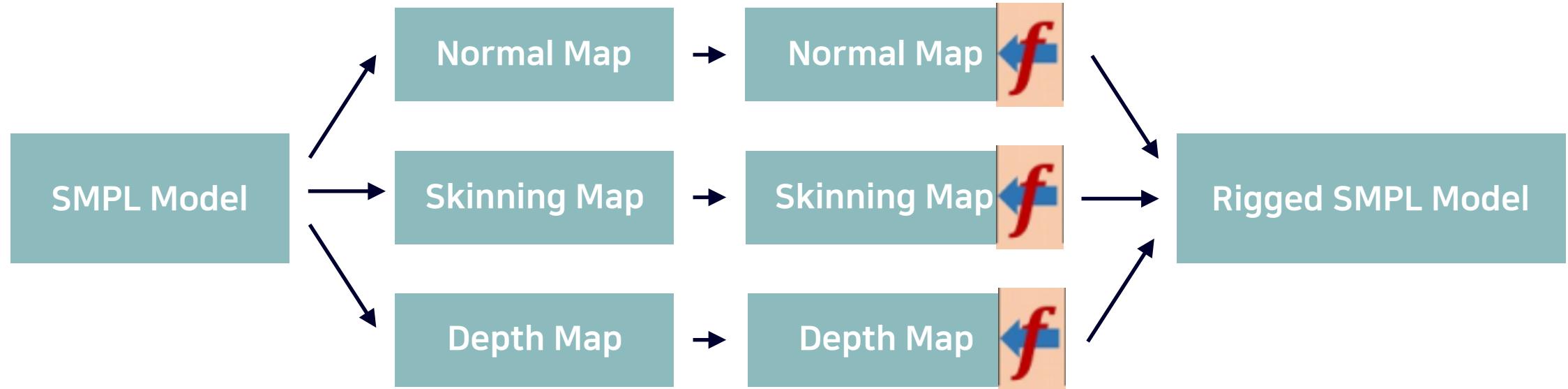
5) Normal map, Skinning map, Depth map



[6] Bogo, Federica, et al. "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image." European Conference on Computer Vision. Springer, Cham, 2016.
[10] Alldieck, Thiemo, et al. "Detailed human avatars from monocular video." IEEE, 2018.

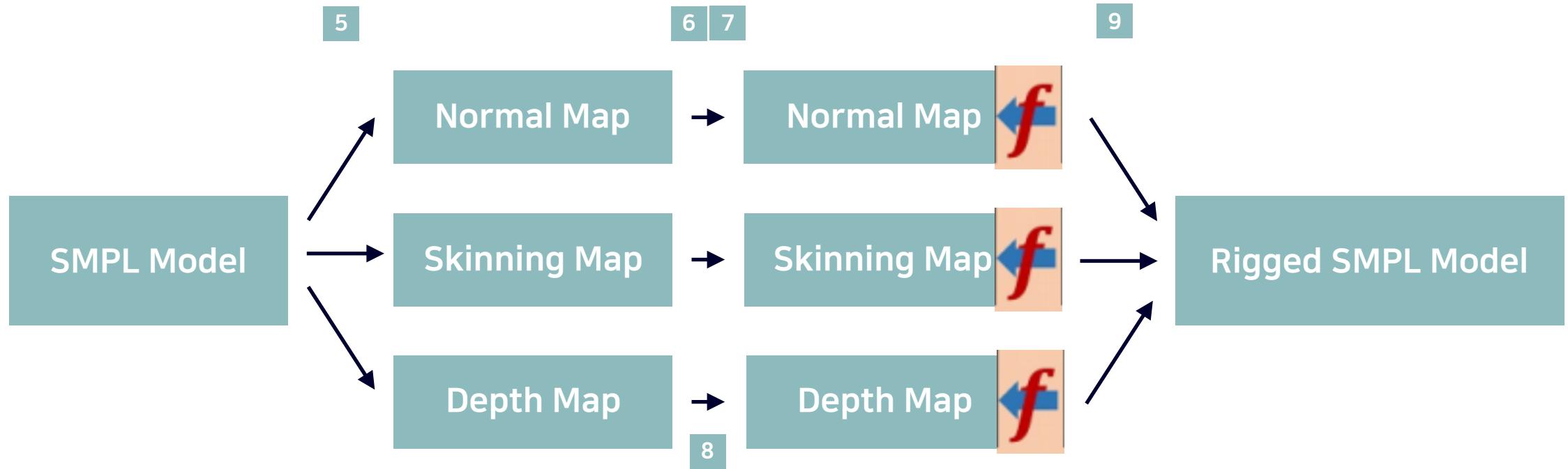
01. Mesh construction & Rigging

5) Normal map, Skinning map, Depth map



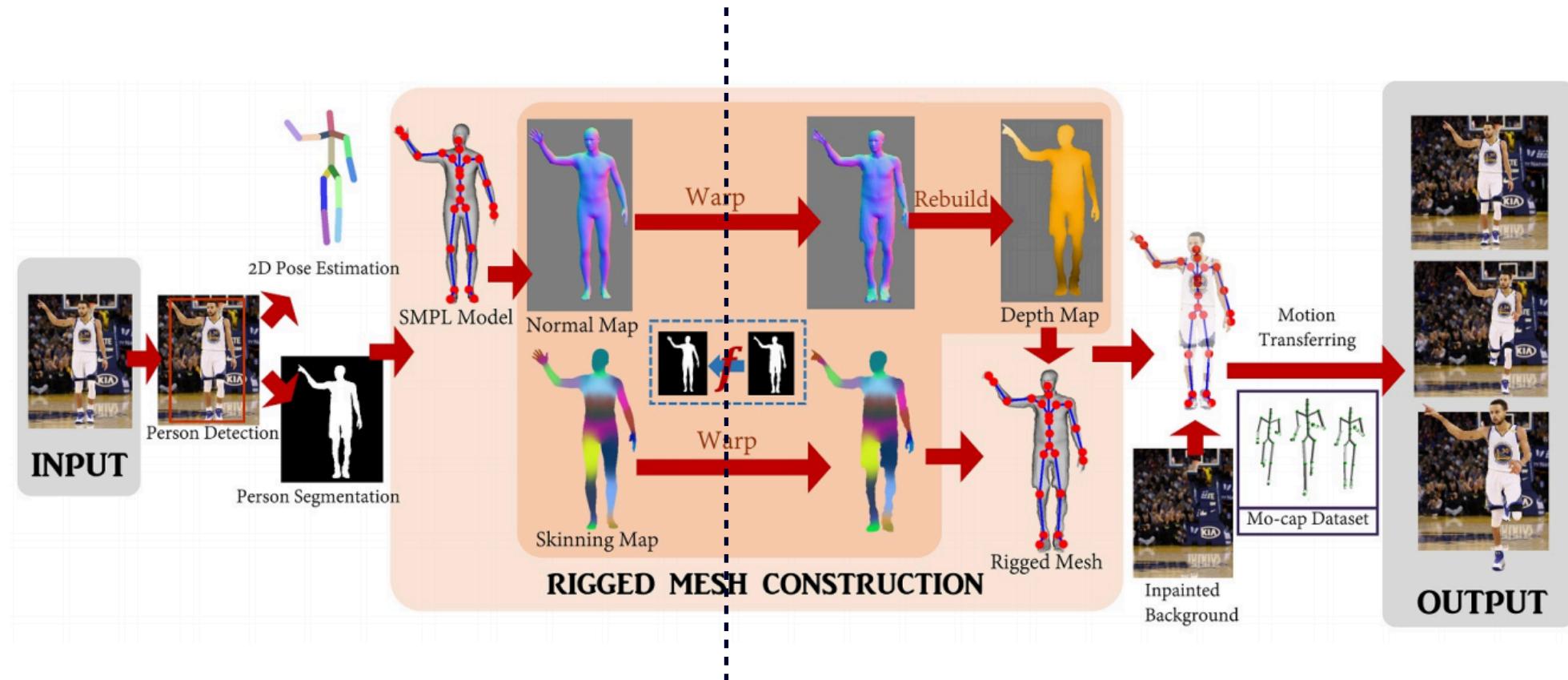
01. Mesh construction & Rigging

5) Normal map, Skinning map, Depth map



01. Mesh construction & Rigging

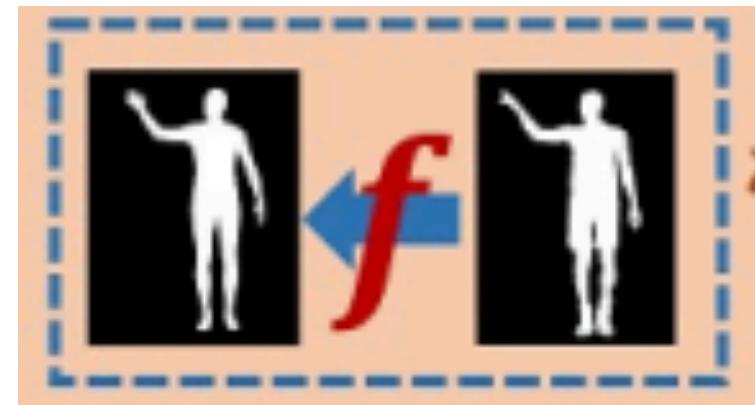
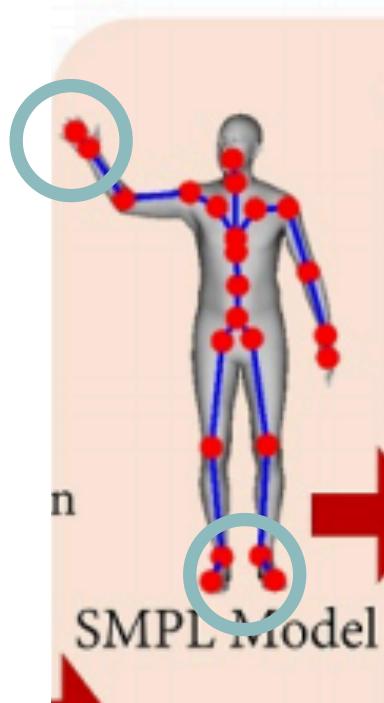
5) Normal map, Skinning map, Depth map



[6] Bogo, Federica, et al. "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image." European Conference on Computer Vision. Springer, Cham, 2016.
[10] Alldieck, Thiemo, et al. "Detailed human avatars from monocular video." IEEE, 2018.

01. Mesh construction & Rigging

6) Inverse Warp

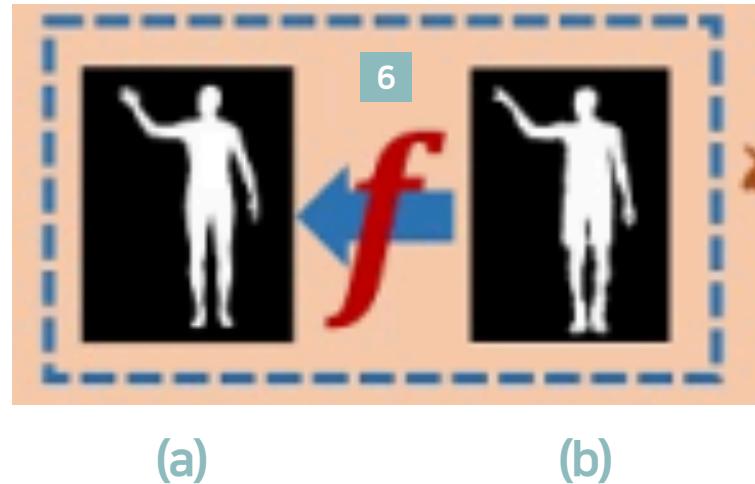


(a)

(b)

01. Mesh construction & Rigging

6) Inverse Warp



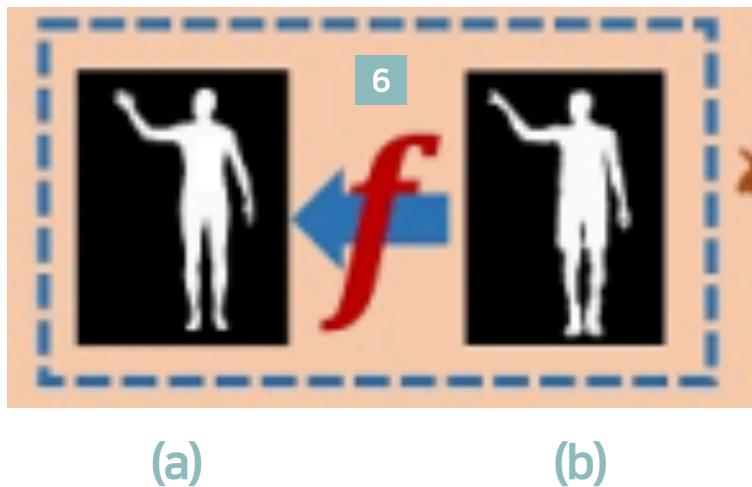
$$x = \sum_{i=0}^{m-1} \lambda_i(x) p_i \quad (6)$$

$$p_i \rightarrow p_{\phi[i]}^{\text{SMPL}}. \quad (7)$$

$$f(x) = \sum_{i=0}^{m-1} \lambda_i(x) p_{\phi[i]}^{\text{SMPL}}. \quad (8)$$

01. Mesh construction & Rigging

7) Boundary matching



$$\arg \min_{\phi[0], \dots, \phi[m-1]} \sum_{i=0}^{m-1} D(p_i, p_{\phi[i]}^{\text{SMPL}}) + T(\phi[i], \phi[i+1]) \quad (9)$$

where

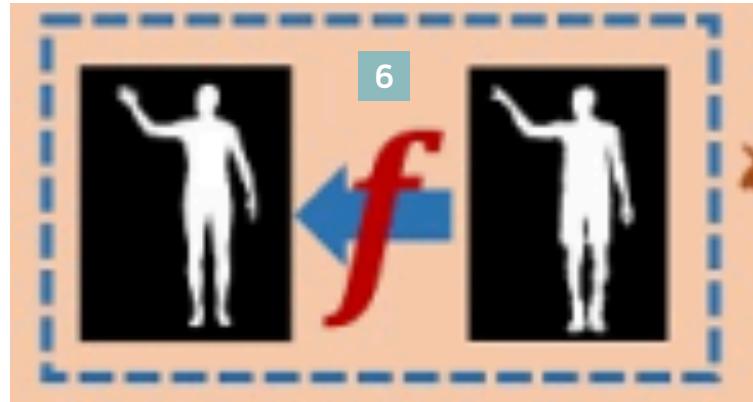
$$D(p_i, p_{\phi[i]}^{\text{SMPL}}) = \|p_i - p_{\phi[i]}^{\text{SMPL}}\|_2 \quad (10)$$

and

$$T(\phi[i], \phi[i+1]) = \begin{cases} 1, & \text{if } 0 \leq \phi[i+1] - \phi[i] \leq \kappa \\ \infty, & \text{otherwise} \end{cases} \quad (11)$$

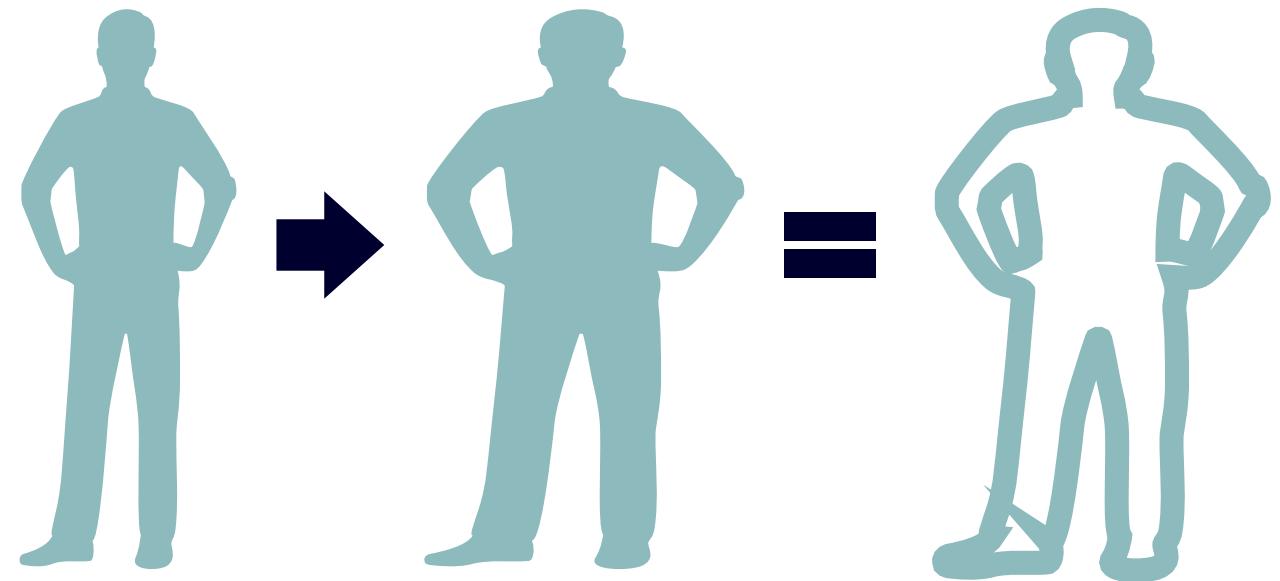
01. Mesh construction & Rigging

7) Boundary matching



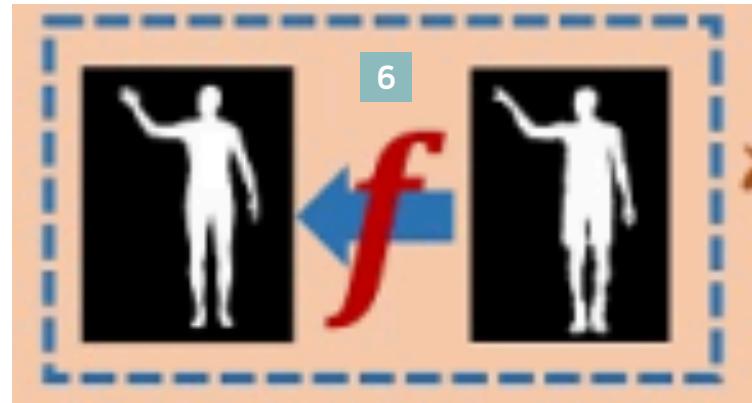
(a)

(b)



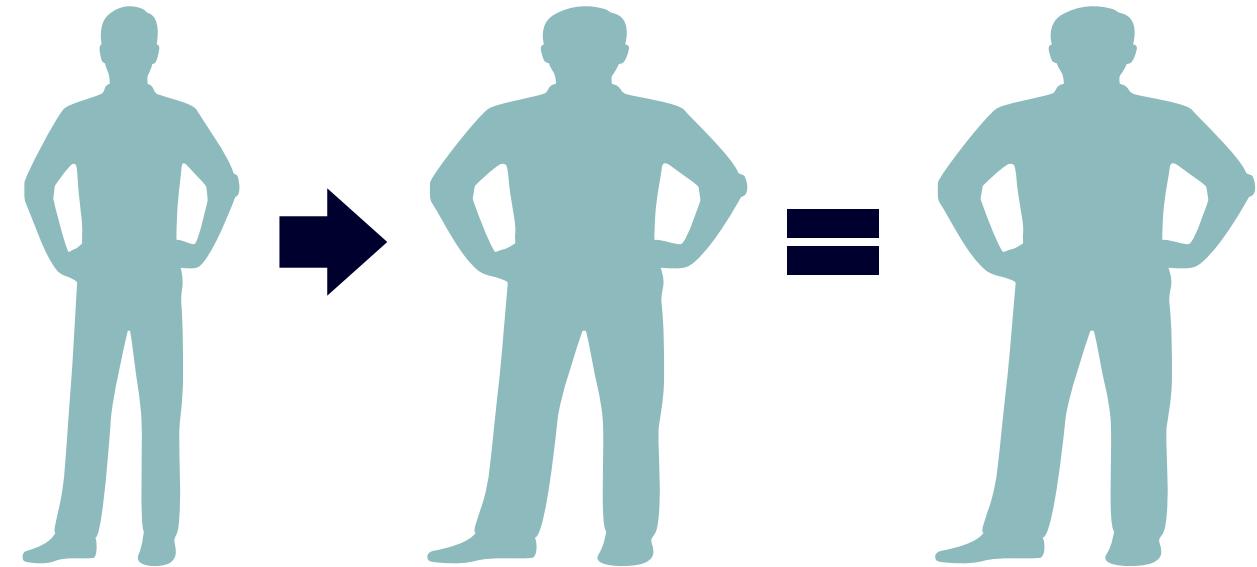
01. Mesh construction & Rigging

7) Boundary matching



(a)

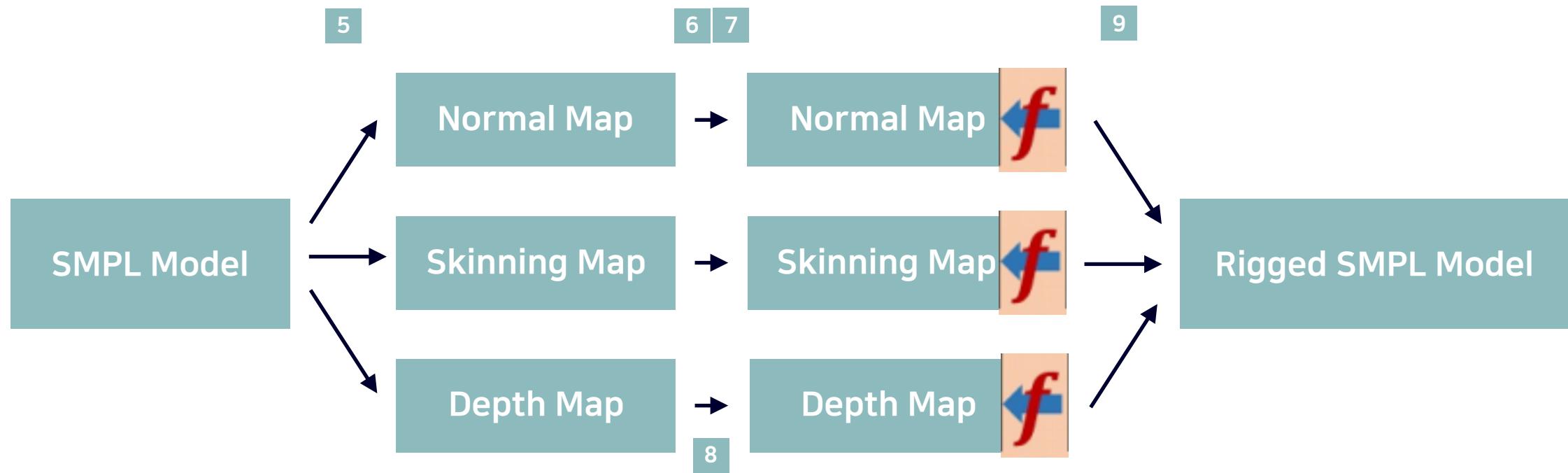
(b)



Hole-filling

01. Mesh construction & Rigging

8) Rebuilding Depth Map



01. Mesh construction & Rigging

8) Rebuilding Depth Map

$$S(x) = S_{\text{SMPL}}(f(x)) \quad (1)$$

$$Z_{\partial S}(x \in \partial S) = Z_{\text{SMPL}}(f(x)) \quad (2)$$

$$N(x) = N_{\text{SMPL}}(f(x)) \quad (3)$$

$$Z(x) = \text{Integrate}[N; Z_{\partial S}] \quad (4)$$

$$W(x) = W_{\text{SMPL}}(f(x)) \quad (5)$$



(a) SMPL



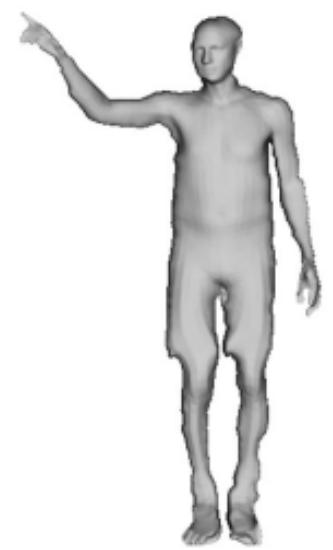
(b) Warp depth



(c) Warp normal
and then integrate

01. Mesh construction & Rigging

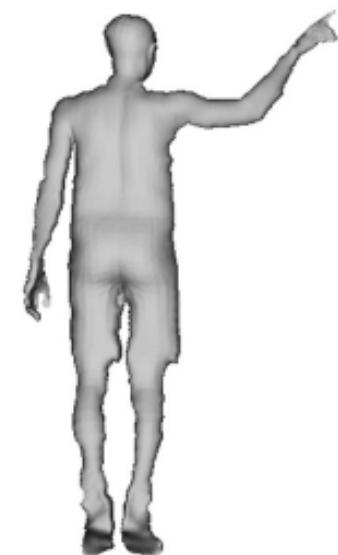
9) Rigged Mesh



(a) front mesh



(b) side view



(c) back mesh

01. Mesh construction & Rigging

10) Inpainted

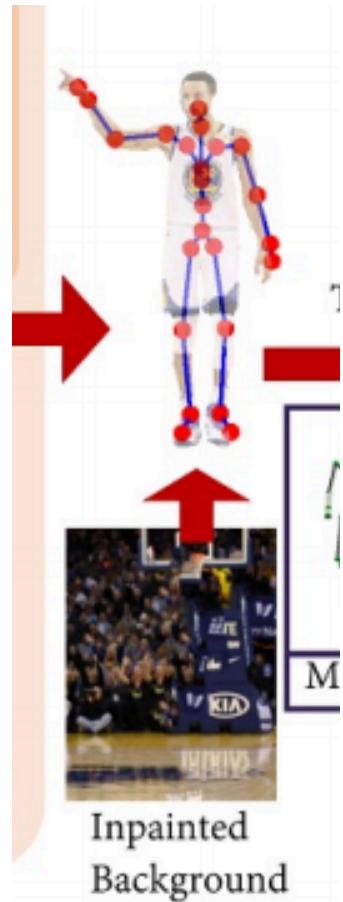
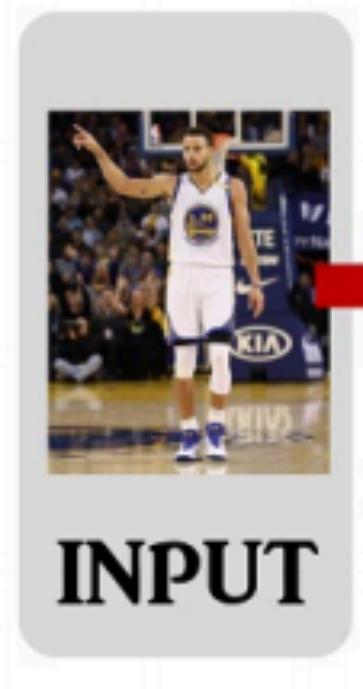


Figure 1: Structural image editing. Left to right: (a) the original image; (b) a hole is marked (magenta) and we use line constraints (red/green/blue) to improve the continuity of the roofline; (c) the hole is filled in; (d) user-supplied line constraints for retargeting; (e) retargeting using constraints eliminates two columns automatically; and (f) user translates the roof upward using reshuffling.

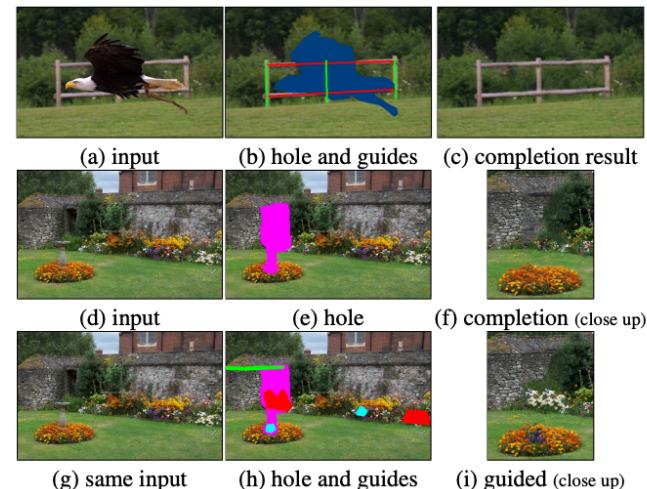


Figure 4: Two examples of guided image completion. The bird is removed from input (a). The user marks the completion region and labels constraints on the search in (b), producing the output (c) in a few seconds. The flowers are removed from input (d), with a user-provided mask (e), resulting in output (f). Starting with the same input (g), the user marks constraints on the flowers and roofline (h), producing an output (i) with modified flower color and roofline.

01. Mesh construction & Rigging

11) Motion transferring

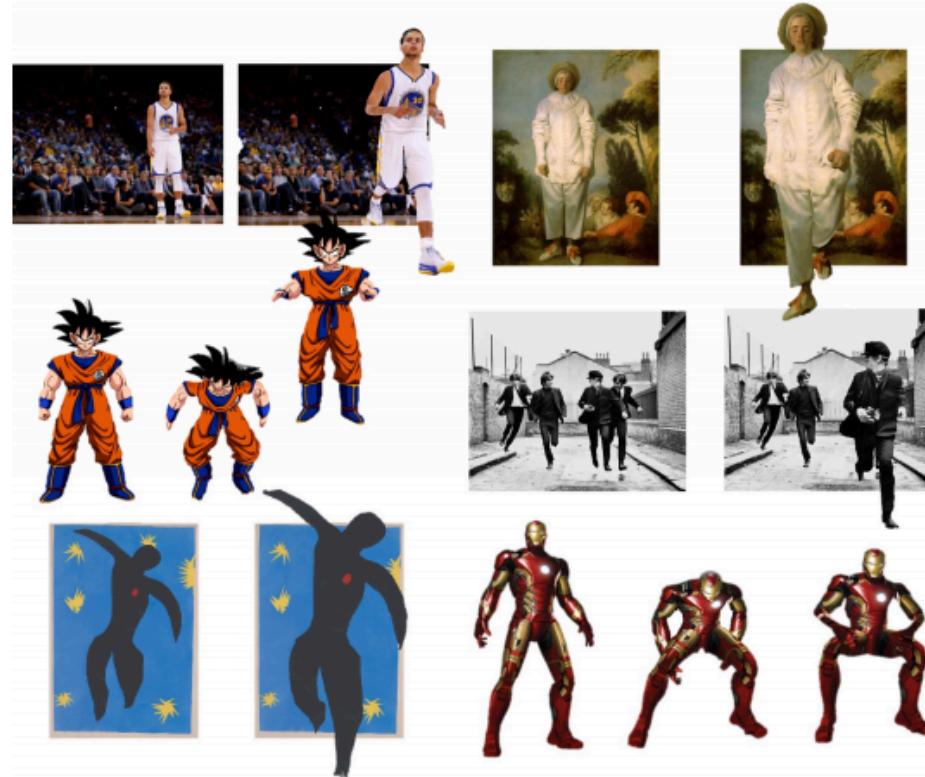
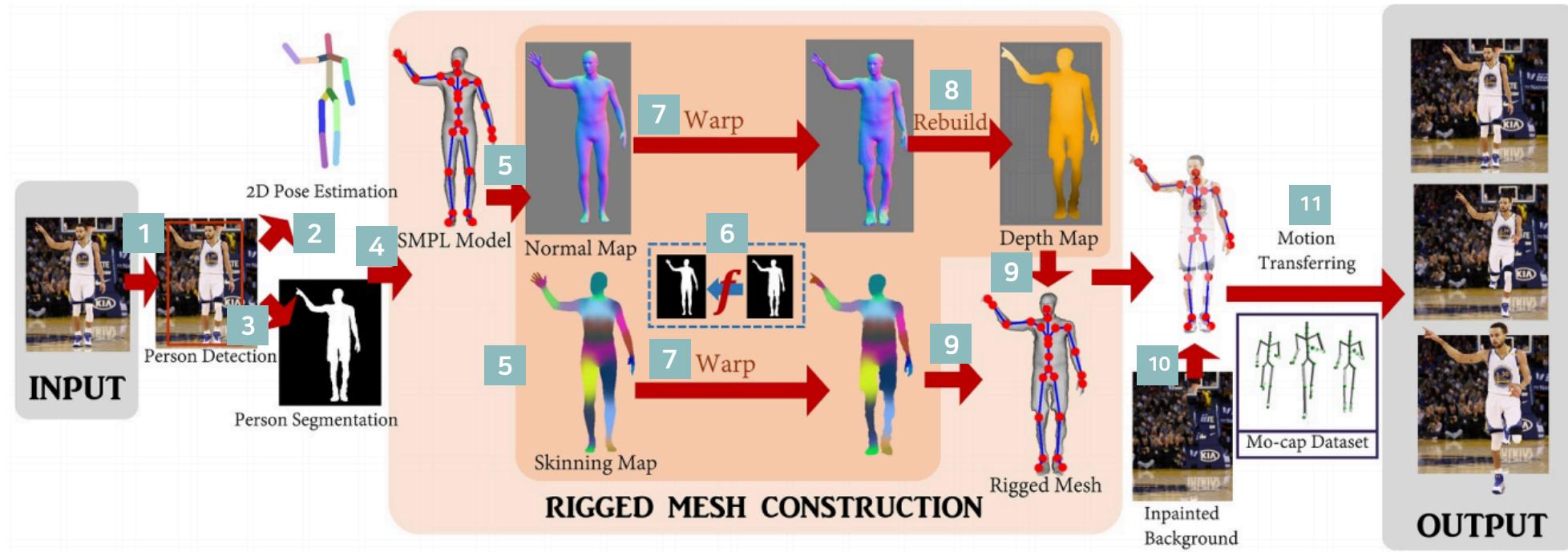


Figure 7: Six animation results. The input is always on left.
Photo credits: [3, 9, 2, 1, 8, 6]

<https://youtu.be/G63goXc5MyU>

01. Mesh construction & Rigging

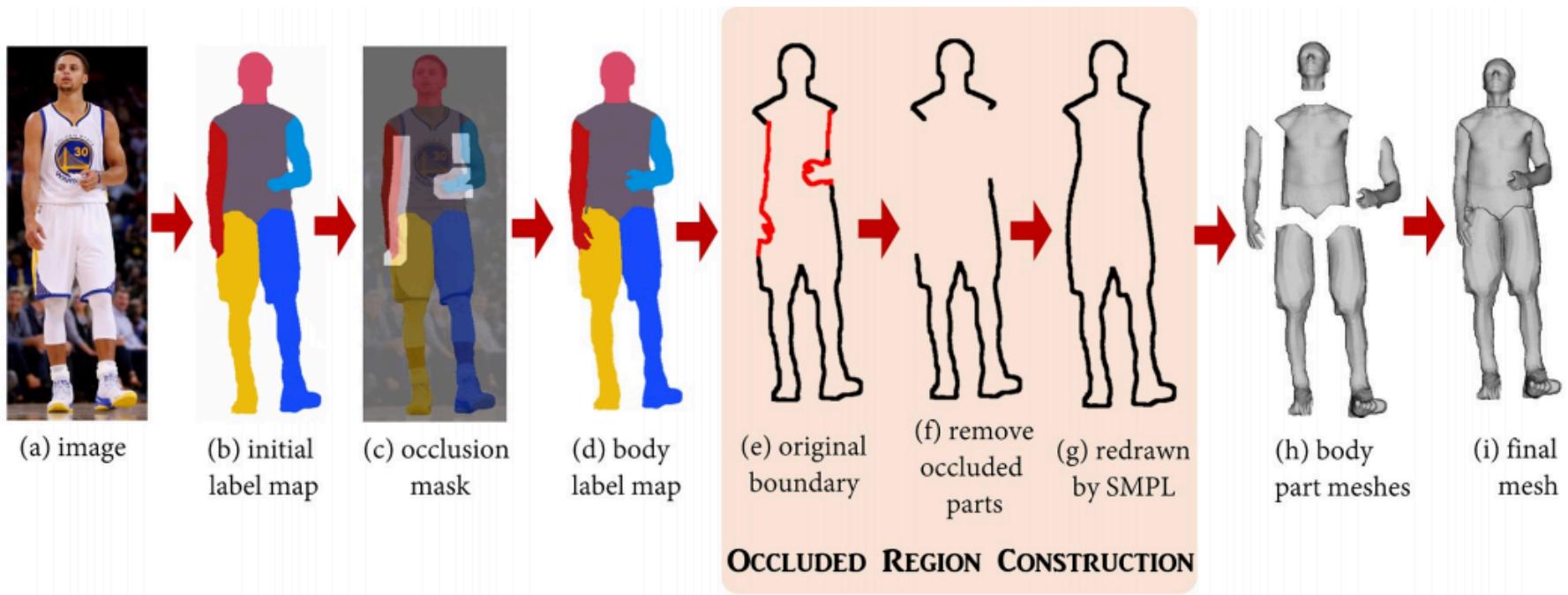
Review



2. Self-occlusion

02. Self-occlusion

self-occlusion?



02. Self-occlusion

(1) Body label map?

(1) estimate an initial label map $L(\text{init})$ for each pixel x to be as similar as possible to $L(\text{SMPL})$

(2) refine $L(\text{init})$ at occlusion boundaries

02. Self-occlusion

(1) Body label map - initial body labeling



$$\min_{L_{\text{init}}} \sum_{p \in S} U(L_{\text{init}}(p)) + \gamma \sum_{p \in S, q \in \mathcal{N}(p) \cap S} V(L_{\text{init}}(p), L_{\text{init}}(q)) \quad (12)$$

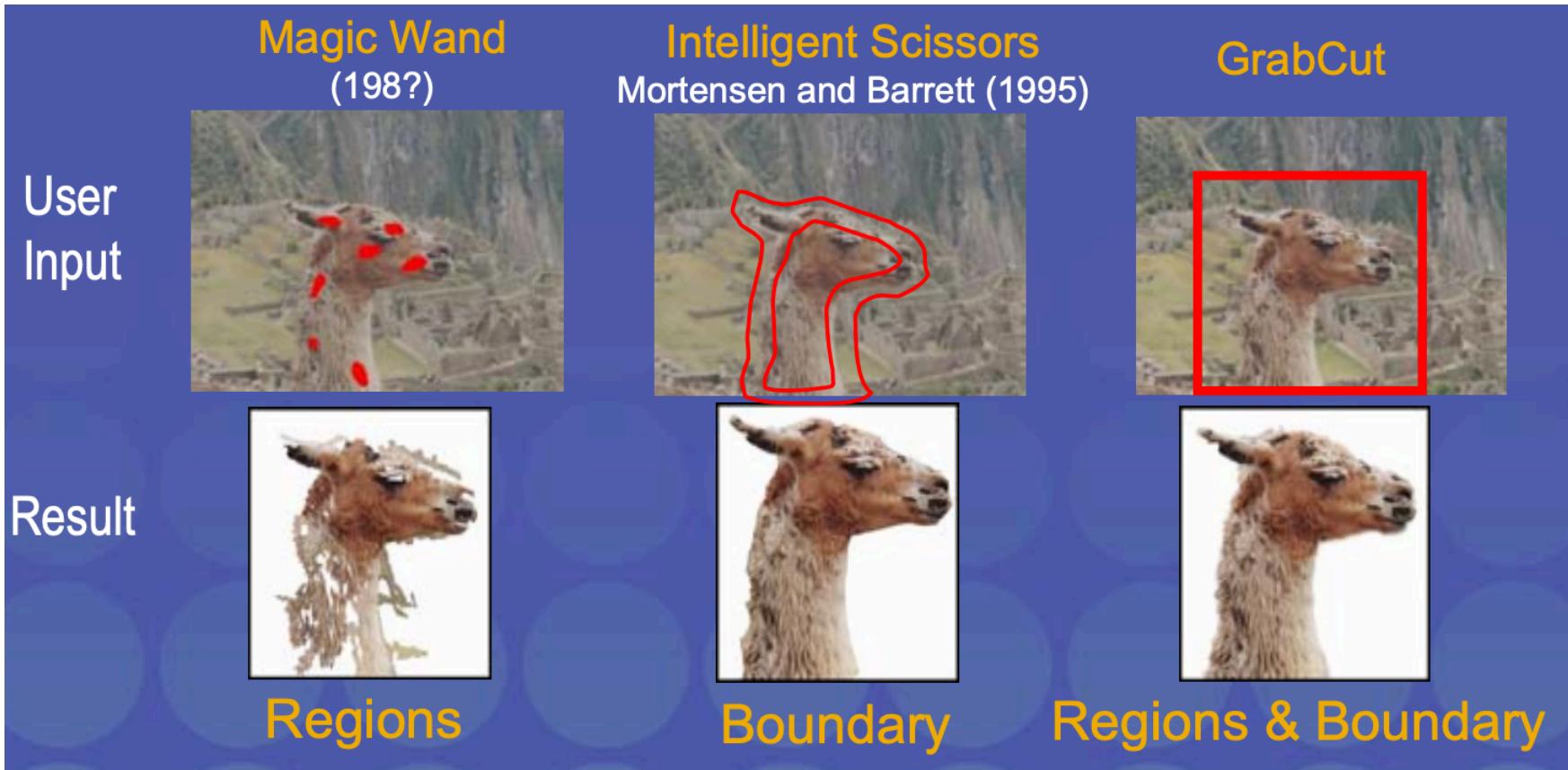
where

$$U(L_{\text{init}}(p)) = \min_{r | L_{\text{SMPL}}(r) = L(p)} \|p - r\|_2 \quad (13)$$

$$V(L_{\text{init}}(p), L_{\text{init}}(q)) = \begin{cases} 1 & \text{if } L_{\text{init}}(p) \neq L_{\text{init}}(q) \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

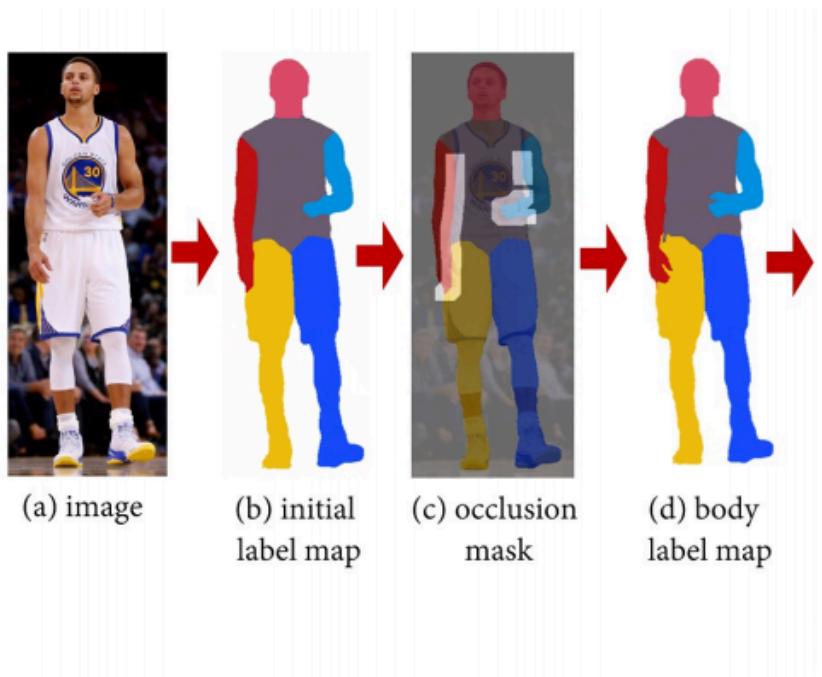
02. Self-occlusion

(1) Body label map - refined body labeling



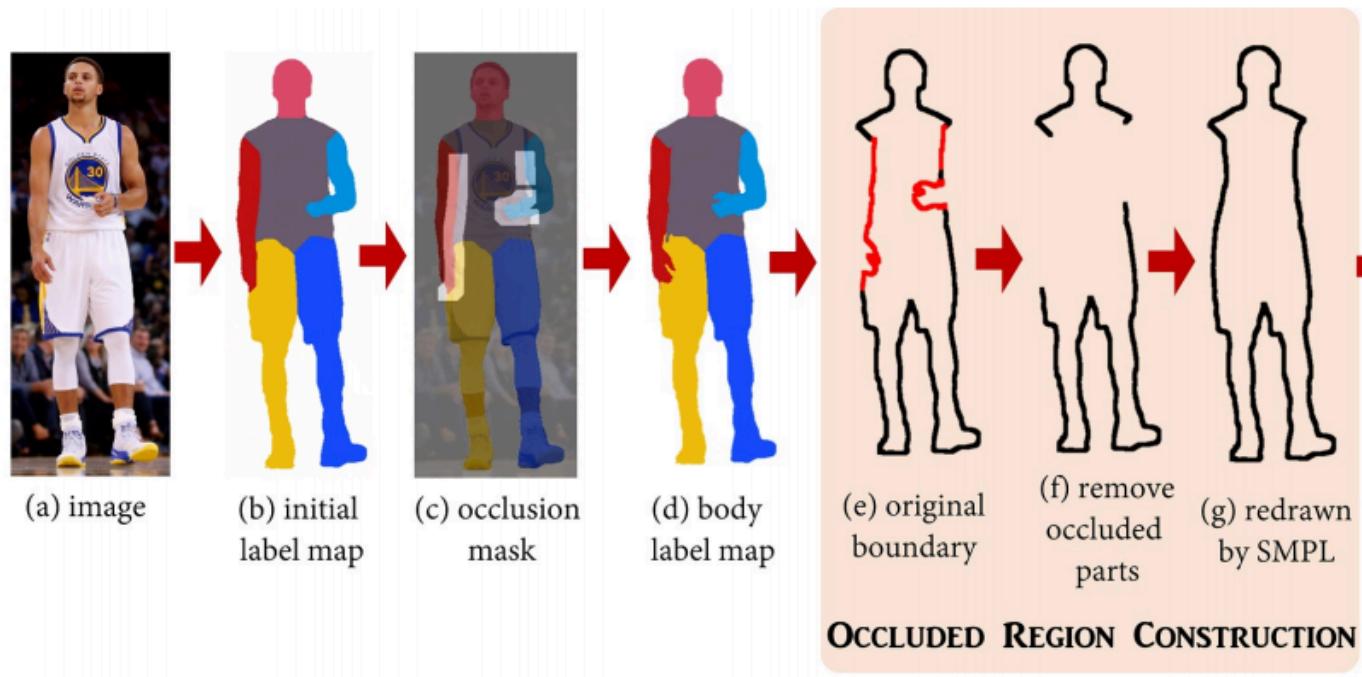
02. Self-occlusion

(1) Body label map - result



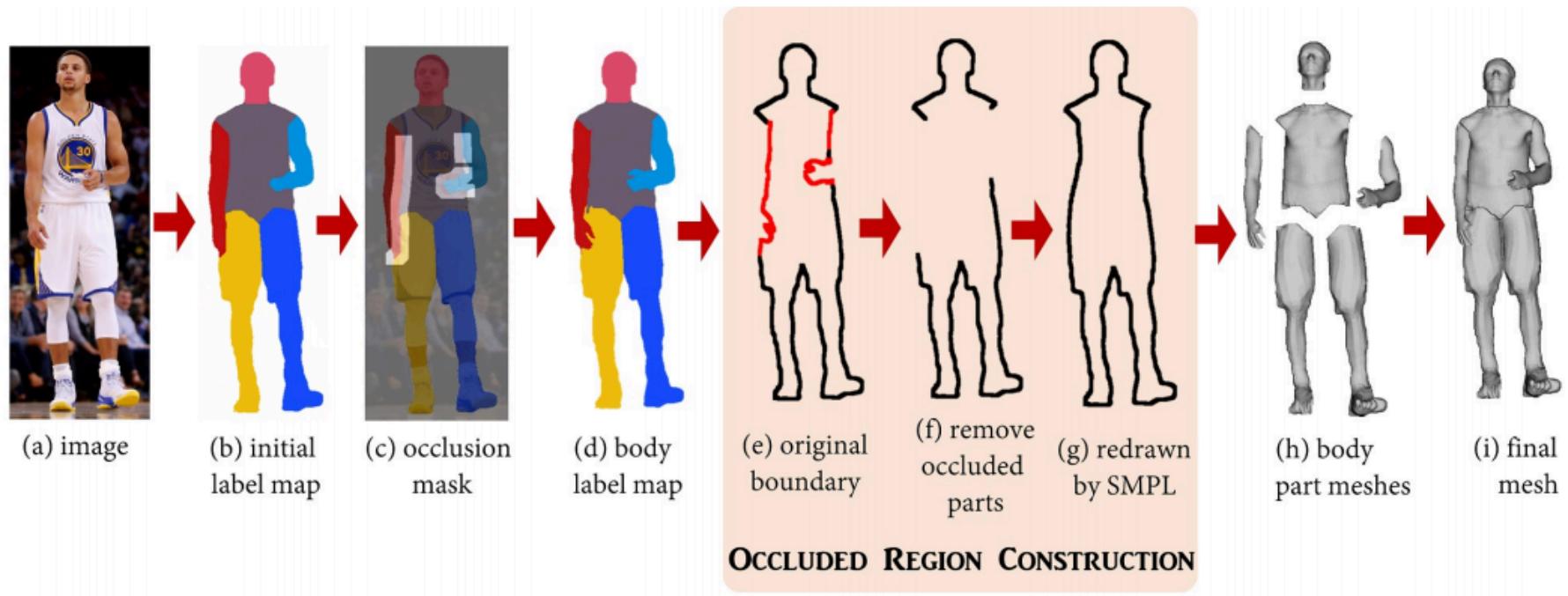
02. Self-occlusion

(2) Occluded region construction



02. Self-occlusion

(3) Mesh construction



02. Self-occlusion

review

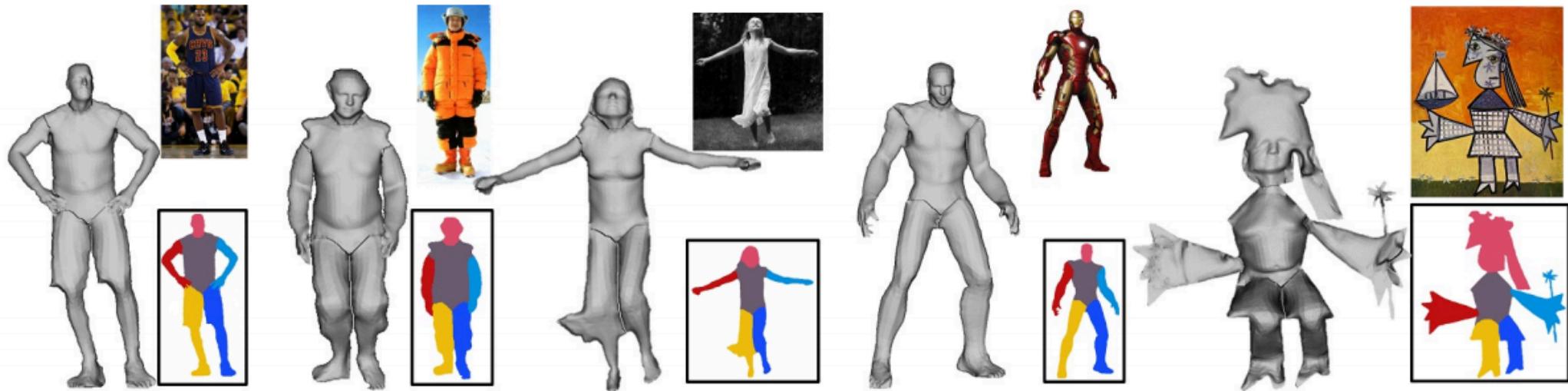


Figure 6: Examples of body label maps and meshes (input photos are put on top right corner). *Photo credits: [3, 5, 4, 6, 8]*

3. Final Steps

03. Final Steps

1) Head pose correction

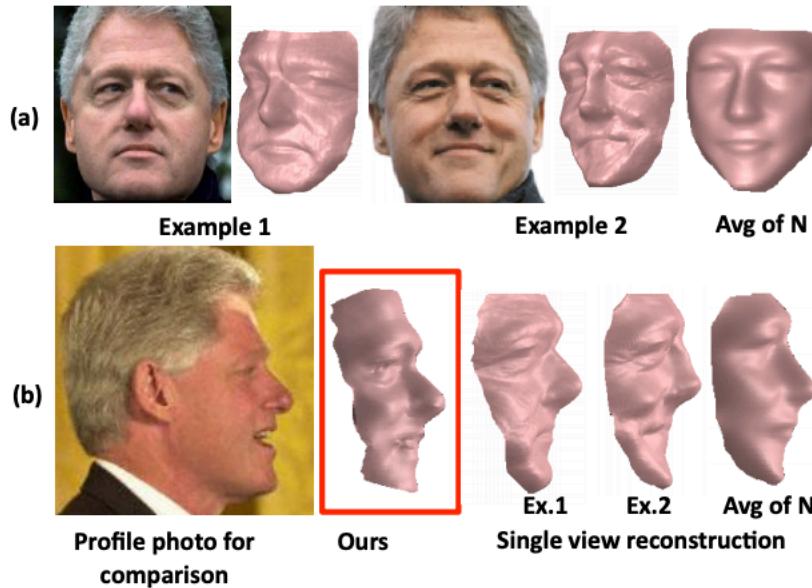


Figure 9. Comparison to single view reconstructions provided by [11]. (a) Two photos of Bill Clinton, the shape reconstruction using the single view method (from each of the images separately), and average of single view reconstructions from all the images in the set. (b) Photo of Clinton's profile, and profile renderings of our reconstruction, two single view reconstructions and average of all the single view reconstructions.

03. Final Steps

2) Texturing

- a. paste a mirrored copy of the front texture onto the back**
- b. inpaint with optional user guidance & poison blending**

03. Final Steps

2) Texturing

a. paste a mirrored copy of the front texture onto the back



03. Final Steps

2) Texturing

b. inpaint with optional user guidance & poison blending

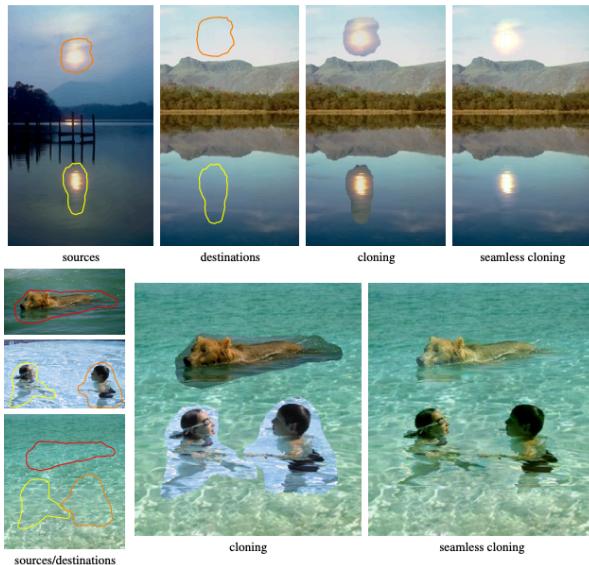


Figure 3: **Insertion**. The power of the method is fully expressed when inserting objects with complex outlines into a new background. Because of the drastic differences between the source and the destination, standard image cloning cannot be used in this case.

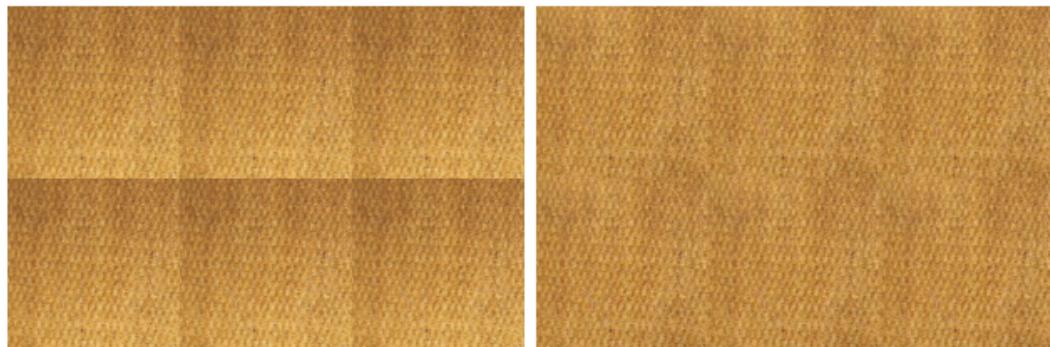


Figure 12: **Seamless tiling**. Setting periodic boundary values on the border of a rectangular region before integrating with the Poisson solver yields a tileable image.

04. Results and Discussion

Experience



(a) input photo (b) Hornung et al. (c) ours – 3D demonstration

103 응시자
86% 선호

Figure 10: Comparison result with [26]: (a) input photo; (b) animation method proposed in [26]; (c) 3D demonstration using our method, which is not possible in [26]. *Photo credit: Hornung et al.*

04. Results and Discussion

Limitations

- (1) Shadows and reflections are currently not modeled by our method and thus won't move with the animation
- (2) The shape may look unrealistic
- (3) Our method accounts for self-occlusions when arms partially occlude the head, torso, or legs
- (4) Person detection and segmentation, pose detection and body labeling can fail, requiring manual corrections
- (5) We have opted for simple texture inpainting for occluded body parts, with some user interaction if needed

05. Q

and A