Supplemental Material: Photo Wake-Up

Chung-Yi Weng, Brian Curless, Ira Kemelmacher-Shlizerman

UW Reality Lab
Paul G. Allen School of Computer Science and Engineering
University of Washington

1. Algorithmic details:

Below we describe additional algorithmic details. Please do check out the supplementary video for results[1].

Hole-filling: In practice, holes may arise when warping by $f(x)$, i.e., small regions in which $f(x) \notin S_{\text{SMPL}}$. An example mesh rebuilt from an ouput depth map with holes is illustrated in Fig. 1(a).

To fill these holes, we again apply mean value coordinates, now for smooth inpainting of normals and skinning weights. For each hole H in normal map N, we collect the points on the hole boundary ∂H, compute mean-value coordinates of the points $x \in H$ in terms of the boundary points, and then interpolate the boundary normals using these coordinates. We do the same for the skinning map W, interpolating the vector skinning weights around ∂H.

A mesh result before and after hole-filling is shown in Fig. 1.

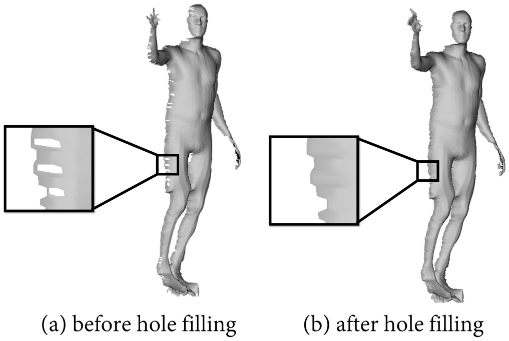
(a) before hole filling (b) after hole filling

Figure 1: Hole-filling results. Holes result can arise when warping between silhouettes. Here we visualize the result of holes in the depth map (a), which we then smoothly fill (b).

Face Alignment: The generic body shape model includes a head model which is fitted during the process just like the body, but typically the optimization fails to predict correct head pose. Incorrect head pose creates strong artifacts when the shape is textured and viewed from the sides. Ultimately, for the animation to include correct facial shape, single view face modeling techniques should be applied, e.g., [2, 5, 9], combined with head modeling techniques. Modeling the full head [7], and from a single image [4] is an active research area, which is beyond the scope of this paper. However, even just aligning the head with face-specific techniques improves our animation results significantly. We align the SMPL head estimation with the face in the photo as follows.

We begin by estimating the head pose and then warp the transformed head mesh according to the image. Head pose is estimated as in [6]. Example result after head correction is shown in Fig. 2(c).

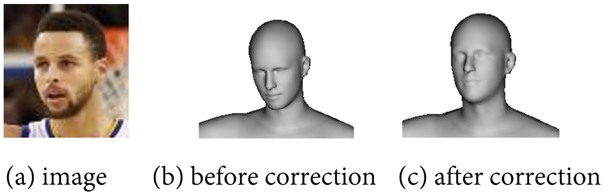
(a) image (b) before correction (c) after correction

Figure 2: In the result of SMPL optimization the head pose is usually incorrect (b), thus we further align the head to detected face fiducials and estimate head pose. The correction result is shown in (c).

Given the corrected head pose, we generate the full depth map Z as before. We warp Z in the head part to match the fiducials to ensure good texturing. We treat 7 fiducial points F (corners of two eyes, nose, and corners of mouth) in the image and boundary points ∂B of the head as anchors. We then project the corresponding 3D fiducial points Q on the SMPL mesh into the depth map Z_{SMPL} and apply mean-value coordinates to find their corresponding locations, F', in the output. Finally, we warp the depth map Z within the head part to map F' to F while keeping the boundary ∂B fixed, via moving least squares transform [8]. We chose MLS due to its good global smoothness property, no need for triangulation, and closed-form solution.

[1] https://youtu.be/G63goXc5MyU

In case of abstract art or other types of photos where the face and fiducials are not detected, the face-specific treatment process is skipped, and the projected generic mesh is warped to the silhouette just as in the body case.

Texturing: Our final step is to texture the reconstructed 3D model. To texture the front mesh (before stitching to the back mesh), we can simply assign colors from the input image to the corresponding vertices back-projected depth map. Due to small errors in person segmentation as well as mixed foreground-background pixels at the silhouette, discolorations may appear on a narrow band near the boundary of the mesh (Fig. 3(a)). These errors may be addressable with more sophisticated segmentation refinement and matting. Instead, we simply erode S to form S' and then replace the color of each pixel in $S \setminus S'$ with the color of the nearest pixel in S' (Fig. 3(a)).

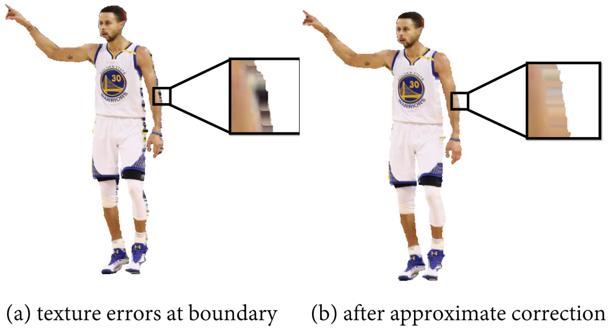
(a) texture errors at boundary (b) after approximate correction

Figure 3: When texturing a mesh, errors arise around the silhouette boundary. We reduce the artifacts by replacing them with the colors nearest neighbor pixels well within the silhouette.

If a frontal region is occluded – e.g., hand in front of the torso – we apply the patch-match algorithm [1] to inpaint the region.

Texturing the back of the body is more difficult, as we have no direct observation of it. One approach is to simply mirror the front texture onto the back. This mirrored-texturing produces reasonable results in some cases (e.g., arms), but undesirable results in others (face appears on the back of the head). To address this problem, we allow the user to choose between mirrored texturing or label-driven texture synthesis – "texture-by-numbers" – on a part-by-part basis. Fig. 4 illustrates the latter approach. Starting from the original body part label map, the user can apply new color labels to the source (frontal) image, and optionally to the back image. We then synthesize texture for the back, restricted to draw from regions with the same label. When texture synthesis does not produce a satisfactory result, the user can opt instead to revert to mirrored-texturing.

Finally we apply Poisson blending[3] to back texture when stitching it with the front texture.

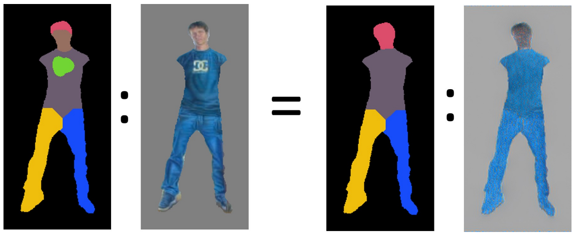

Figure 4: Synthesizing back texture. We transform the back texture construction into a texture-by-numbers problem. We first modify the body label map by labeling undesired region with different colors (i.e., face and shirt logo) to create the source label map. In this example, we then use the the original body label map as the target label map for the back; thus, the constrained texture synthesis will not use pixels covered by the new labels when creating the back texture, so that the face and logo do not appear on the back.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28:24–1, 2009. 2

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 1

[3] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski. Coordinates for instant image cloning. In *ACM Transactions on Graphics (TOG)*, volume 28, page 67. ACM, 2009. 2

[4] L. Hu, S. Saito, L. Wei, K. Nagano, J. Seo, J. Fursund, I. Sadeghi, C. Sun, Y.-C. Chen, and H. Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6):195:1–195:14, Nov. 2017. 1

[5] I. Kemelmacher-Shlizerman and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):394–405, 2011. 1

[6] I. Kemelmacher-Shlizerman and S. M. Seitz. Face reconstruction in the wild. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1746–1753. IEEE, 2011. 1

[7] S. Liang, L. G. Shapiro, and I. Kemelmacher-Shlizerman. Head reconstruction from internet photos. In *European Conference on Computer Vision*, pages 360–374. Springer, 2016. 1

[8] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *ACM transactions on graphics (TOG)*, volume 25, pages 533–540. ACM, 2006. 1

[9] A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1502. IEEE, 2017. 1