Photo Wake Up

Presented By:

Yuval Reshef

Tal Haklay

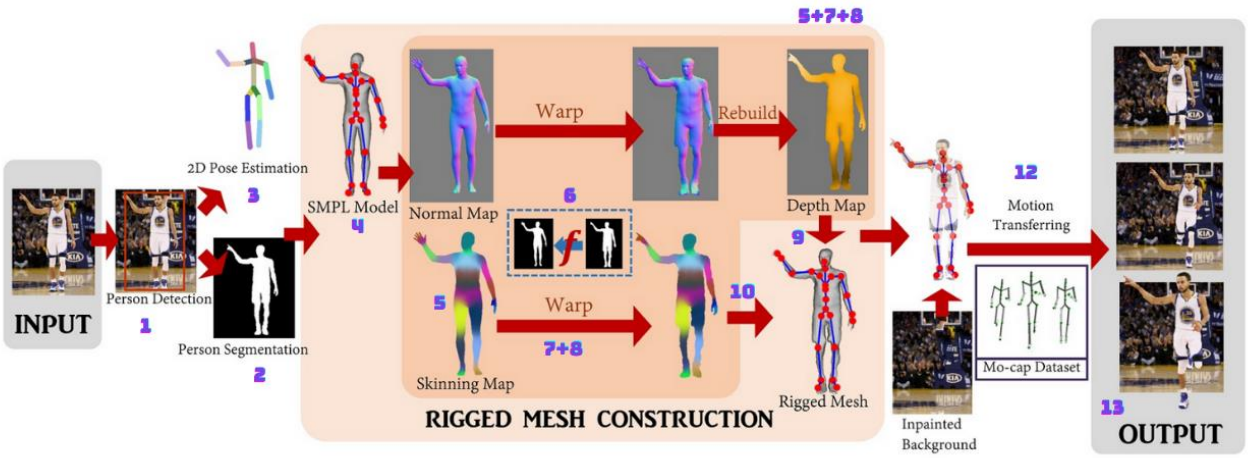
Asaf Buchnick

Introduction

We shall present the method proposed in *"Photo Wake-Up: 3D Character Animation from a Single Photo"* (*Weng Et al.*), and show our implementation for it, based on the steps provided in the paper.

Our pipeline takes a single photo of a human character as an input, and outputs an animated version of it, using a specific animation we decided to focus on.

Our project's main contribution is using new and improved libraries to implement the pipeline described in the paper.



Pipeline Implementation

We will describe the general outline of the pipeline steps, as described in the image attached above.

1. We start by identifying the main human character in the image, using Mask R-CNN model with a ResNet-50-FPN backbone (“Mask R-CNN”, He Et al.). This step outputs bounding boxes and segmentation masks for every human character that can be found in the input photo. Next, we take the mask of the most significant human character in the photo by selecting the one with the largest corresponding bounding box.

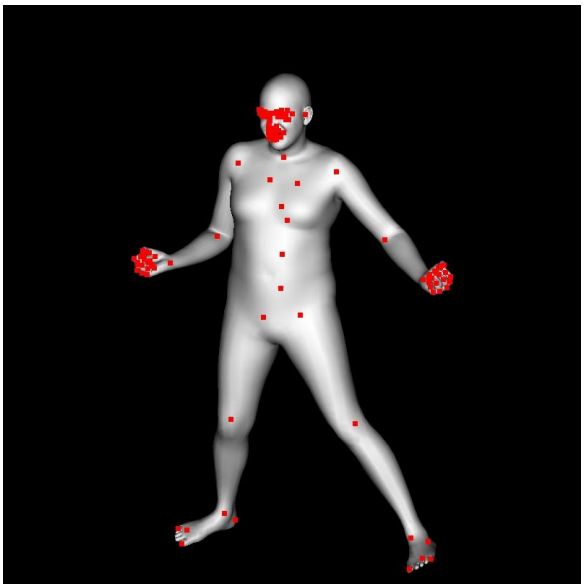
2. Once we have the segmentation mask of the most significant human character, we refine it using "CascadePSP network" (“CascadePSP: Toward Class-Agnostic and Very High-Resolution Segmentation via Global and Local Refinement”, Cheng, Chung Et al.). This significantly improves the segmentation mask compared to the method suggested in the original

paper.

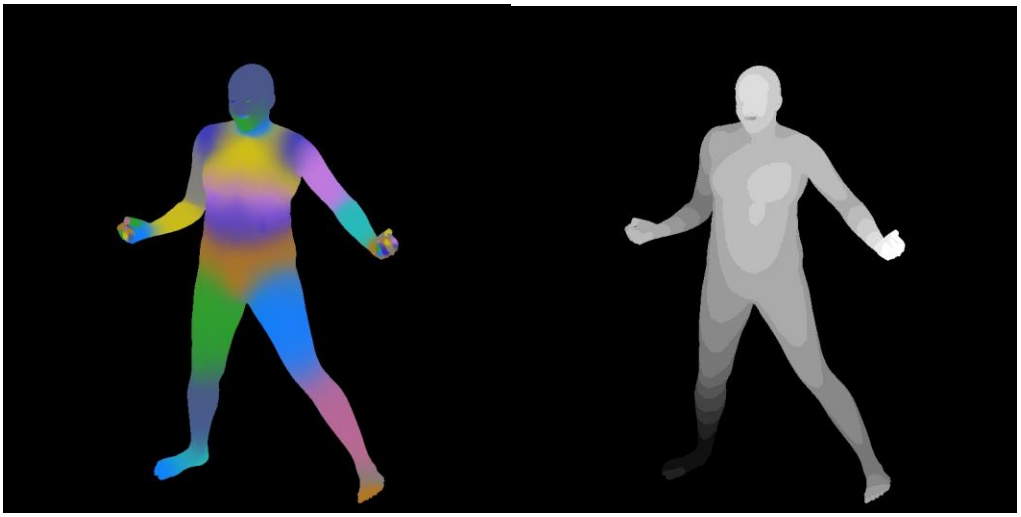
3. Next, we compute the 2D pose estimation, based on the original input photo and OpenPose ("OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", Cao et al.).

4. Based on the results from steps 2 and 3, we run SMPLify-X ("Expressive Body Capture: 3D Hands, Face, and Body from a Single Image", Pavlakos et al.) to obtain the pose and parameters of the SMPLX model of the human character. This is another improvement we added on top of the original paper which used the SMPL model, in order to have a more expressive human model, which also includes fully articulated hands and an expressive

face.



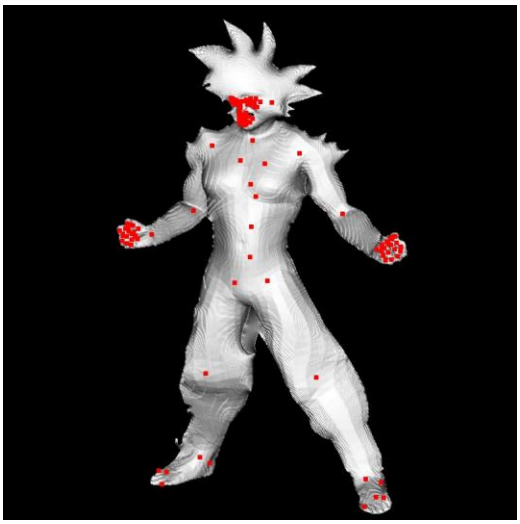
5. The SMPLX model mesh is then used to render skinning and depth maps for both the front and back sides of the SMPLX model. In addition, we retrieve the SMPLX model segmentation mask, the camera parameters of the image (for rendering) and the joints locations of the posed model.



6. To determine the warp function, we use Dynamic Programming to solve the optimization problem that is proposed in the paper. The solution is a mapping between the points on the silhouette of the person and the SMPLX model silhouette. Then we calculate the warp function using the mean-value coordinates of the silhouette mapping described above.

7. We apply the warp function on the front and back depth maps and skinning map from step 5 to obtain the maps for the input character.

8. For better results, we then use hole-filling technique showed on supplementary material of the Photo Wake Up paper, on the depth maps and the skinning map from step 7.
9. A rigid mesh is constructed based on the depth maps from the previous step and the segmentation mask of the input character by creating a vertex for each pixel and connecting faces. The vertices are then projected back to global coordinates using the inverse of the projection matrix used for rendering.



10. We take the generated mesh, and sample skinning weights for each vertex from the skinning map by projecting them to the image.

11. We use the skinning weights, SMPLX joint locations and SMPLX 3D pose data to transform the generated mesh to a T pose to prepare it for animation.

12. Finally, using animations from AMASS dataset *("AMASS: Archive of Motion Capture as Surface Shapes", Mahmood Et al.)*, we transform the T-posed mesh to the pose at each frame of the animation by using linear-blend-

skinning and rigid transformations of the vertices.

13. We then render our posed mesh at each frame using the input image as texture (using flat projection of the input image at the original pose of the mesh), and combine all frames to a video.