Photo Wake-Up

3D Character Animation from a Single Photo

Levon Khachatryan

Applied Statistics and Data Science Graduate Program

In recent years, tremendous amount of progress is being made in the field of 3D Machine Learning, which is an interdisciplinary field that fuses computer vision, computer graphics and machine learning. Photo wake-up belongs to that field. Researchers at the University of Washington have developed this technique for animating a human subject (such as walking toward the screen, running, sitting, and jumping) from a single photo. The technique is demonstrated by using a variety of single-image inputs, including photographs, realistic illustrations, cartoon drawings, and abstracted human forms. The output animation can be played as a video, viewed interactively on a monitor, and as an augmented or virtual reality experience, where a user with headset can enjoy the central figure of a photo coming out into the real world.

Figure 1: Given a single photo as input (far left), model creates a 3D animatable version of the subject, which can now walk towards the viewer (middle). The 3D result can be experienced in augmented reality (right); in the result above the user has virtually hung the artwork with a HoloLens headset and can watch the character run out of the painting from different views.

The overall system works as follows (Fig. 2): First apply state-of-the-art algorithms to perform person detection, segmentation, and 2D pose estimation. From the results, devise a method to construct a rigged mesh. Any 3D motion sequence can then be used to animate the rigged mesh. More specifically, **Mask R-CNN** is used for person detection and segmentation. 2D body pose is estimated using **Convolutional pose machines**, and person segmentation is refined using **Dense CRF**. Once the person is segmented out of the photo, **PatchMatch** (a randomized correspondence algorithm for structural image editing) is applied to fill in the regions where the person used to be.

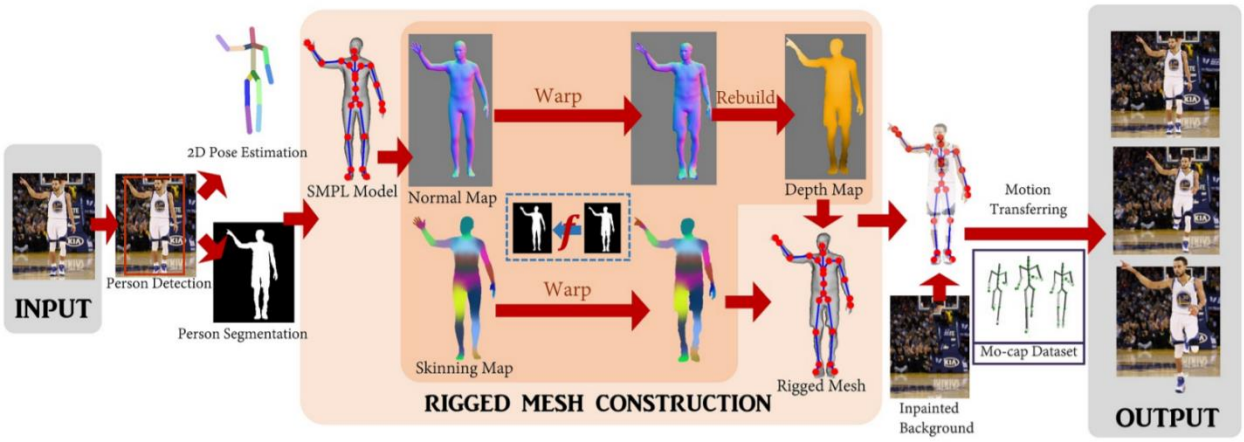

Figure 2: Overview of our method. Given a photo, person detection, 2D pose estimation, and person segmentation, is performed using off-the-shelf algorithms. Then, A SMPL template model is fit to the 2D pose and projected into the image as a normal map and a skinning map. The core of our system is: find a mapping between person's silhouette and the SMPL silhouette, warp the SMPL normal/skinning maps to the output, and build a depth map by integrating the warped normal map. This process is repeated to simulate the model's back view and combine depth and skinning maps to create a complete, rigged 3D mesh. The mesh is further textured, and animated using motion capture sequences on an unpainted background.

An Overview on Techniques for Photo Wake-Up

Mask R-CNN: Used for person detection and segmentation which is based on *Faster R-CNN,* so let's begin by briefly reviewing this detector. *Faster R-CNN* consists of two stages. The first stage, called a *Region Proposal Network (RPN)*, proposes candidate object bounding boxes (we also call them region of interest). The second stage, which is in essence *Fast R-CNN*, extracts features using *RoIPool* from each candidate box and performs classification and bounding-box regression. The features used by both stages can be shared for faster inference.

Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI (region of interest).

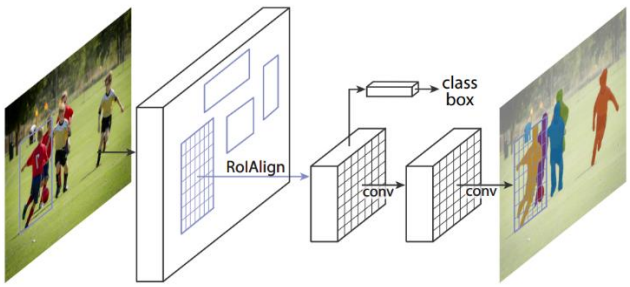

Figure 3. The Mask R-CNN. Key element is RoIAlign which is the main missing piece of Fast/Faster R-CNN.

Dense CRF: As we mentioned above, person segmentation is refined using *dense CRF.* A common approach of pixel-level models for segmentation/detection (as Mask R-CNN does) is to pose problem as *maximum a posteriori* (MAP) inference in a *conditional random field* (CRF) defined over pixels or image patches. The CRF potentials incorporate smoothness terms that maximize label agreement between similar pixels, and can integrate more elaborate terms that model contextual relationships between object classes.

Basic CRF models are composed of unary potentials on individual pixels or image patches and pairwise potentials on neighboring pixels or patches but fully connected (also called dense) CRF establishes pairwise potentials on all pairs of pixels in the image. The pairwise edge potentials are defined by a linear combination of Gaussian kernels in an arbitrary feature space. The algorithm is based on a mean field approximation to the CRF distribution.

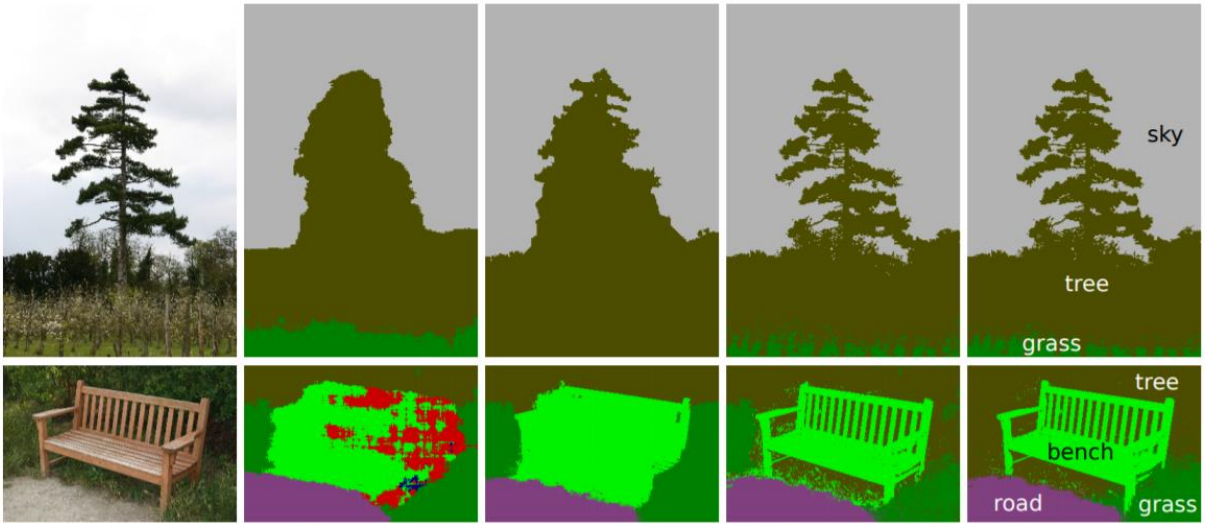

Figure 4: Pixel-level classification with a fully connected CRF. (a) Input image from the MSRC-21 dataset. (b) The response of unary classifiers. (c) Classification produced by the Robust P^n CRF. (d) Classification produced by MCMC inference in a fully connected pixel-level CRF model; the algorithm 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.

PatchMatch: Used to fill in the holes in the person after being segmented. The core of the system is the algorithm for computing patch correspondences. We define a nearest-neighbor field (NNF) as a function $f : A \rightarrow R^2$ of offsets, defined over all possible patch coordinates (locations of patch centers) in image A, for some distance function of two patches D. Given patch coordinate a in image A and its

corresponding nearest neighbor b in image B, $f(a)$ is simply $b - a$. We refer to the values of f as offsets, and they are stored in an array whose dimensions are those of A.

A randomized algorithm for computing an approximate NNF is used. The key insights that motivate this algorithm are that we search in the space of possible offsets, that adjacent offsets search cooperatively, and that even a random offset is likely to be a good guess for many patches over a large image.

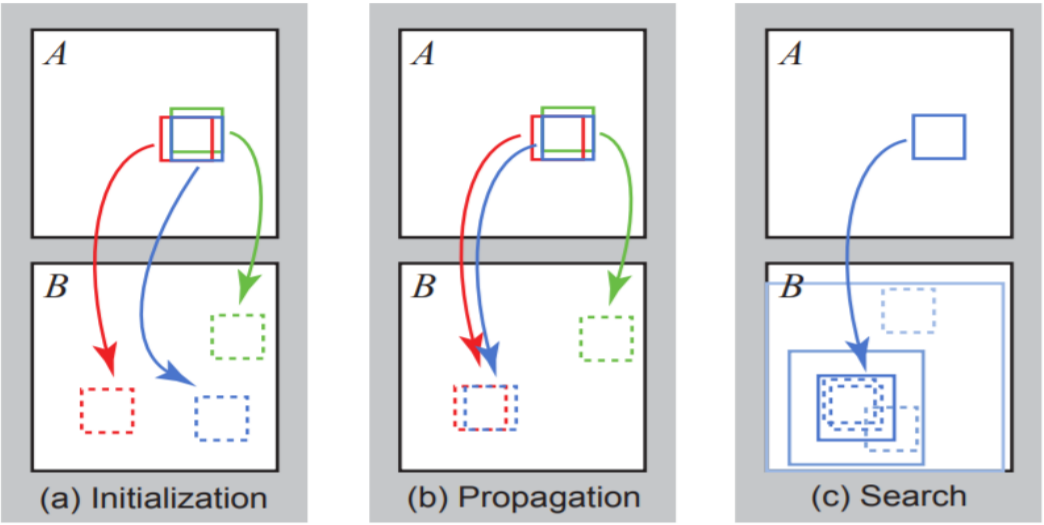

Figure 5: Phases of the randomized nearest neighbor algorithm: (a) patches initially have random assignments; (b) the blue patch checks above/green and left/red neighbors to see if they will improve the blue mapping, propagating good matches; (c) the patch searches randomly for improvements in concentric neighborhoods.

The algorithm has three main components, illustrated in Figure 5. Initially, the nearest-neighbor field is filled with either random offsets or some prior information. Next, an iterative update process is applied to the NNF, in which good patch offsets are propagated to adjacent pixels, followed by random search in the neighborhood of the best offset found so far.

Convolutional Pose Machines (CPMs): Used for the task of articulated pose estimation. CPMs consist of a sequence of convolutional networks that repeatedly produce 2D belief maps for the location of each part. At each stage in a CPM, image features and the belief maps produced by the previous stage are used as input. The belief maps provide the subsequent stage an expressive non-parametric encoding of the spatial uncertainty of location for each part, allowing the CPM to learn rich image-dependent spatial models of the relationships between parts. Instead of explicitly parsing such belief maps either using graphical models or specialized post-processing steps, we learn convolutional networks that directly operate on intermediate belief maps and learn implicit image-dependent spatial models of the relationships between parts. The overall proposed multistage architecture is fully differentiable and therefore can be trained in an end-to-end fashion using backpropagation.

At a particular stage in the CPM, the spatial context of part beliefs provides strong disambiguating cues to a subsequent stage. As a result, each stage of a CPM produces belief maps with increasingly refined estimates for the locations of each part (see Figure 6). In order to capture long range interactions between parts, the design of the network in each stage of our sequential prediction framework is motivated by the goal of achieving a large receptive field on both the image and the belief maps.

Figure 6: A Convolutional Pose Machine consists of a sequence of predictors trained to make dense predictions at each image location. Here we show the increasingly refined estimates for the location of the right elbow in each stage of the sequence. (a) Predicting from local evidence often causes confusion. (b) Multi-part context helps resolve ambiguity. (c) Additional iterations help converge to a certain solution.

Mesh Construction and Rigging:

The key technical idea of Photo Wake-Up method is how to recover an animatable, textured 3D mesh from a single photo to fit the proposed application. Mesh construction and rigging block is responsible for that. It begins by fitting the SMPL (skinned multi-person linear) morphable body model to a photo, including the follow-on method for fitting a shape in 3D to the 2D skeleton (method was published in 2016 and called “automatic estimation of 3D human pose and shape from a single image”). The recovered SMPL model provides an excellent starting point, but it is semi-nude, does not conform to the underlying body shape of the person and, importantly, does not match the clothed silhouette of the person. To that end 2D approach was taken: warp the SMPL silhouette to match the person silhouette in the original image and then apply that warp to projected SMPL normal maps and skinning maps. The resulting normal and skinning maps can be constructed for both front and (imputed) back views and then lifted into 3D, along with the fitted 3D skeleton, to recover a rigged body mesh that exactly agrees with the silhouettes, ready for animation. The center box in Figure 2 illustrates the approach.

Texturing

The 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. 7(a)). These errors may be addressable with more sophisticated segmentation refinement and matting. Instead, we simply erode S (where S is the 2D pose of the person and the person's silhouette mask) to form S' and then replace the color of each pixel in $\frac{S}{S'}$ with the color of the nearest pixel in S' (Fig. 7(a)).

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

Figure 7: 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.

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 are allowed to choose between mirrored texturing or label-driven texture synthesis – "texture-by-numbers" – on a part-by-part basis. Fig. 8 illustrates the latter approach. Starting from the original body part label map, we 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. Finally, we apply Poisson blending to back texture when stitching it with the front texture.

Figure 8: 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 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

Chung-Yi Weng, Brian Curless, Ira Kemelmacher-Shlizerman: Photo Wake-Up: 3D Character Animation from a Single Photo.

T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. PonsMoll: Video based reconstruction of 3d people models.

C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing.

F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black: Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image.

K. He, G. Gkioxari, P. Dollar, and R. Girshick: Mask R-CNN

I. Kemelmacher-Shlizerman and S. M. Seitz: Face reconstruction in the wild.

P. Krahenbuhl and V. Koltun: Efficient inference in fully connected CRFs with gaussian edge potentials.

S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh: Convolutional pose machines.

L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz: Disentangled person image generation.