Dressing a SMPLified Model with Digital, 3D Garments

Jihoon Kim
University of California, Los Angeles
CS 299 - Capstone Project

December 6, 2021

I. Abstract

The recent advances in the deep learning has led to the developments of prediction algorithms based on computer vision. In particular, this paper discusses how to create a seamless pipeline from a single 2D static photo of a person with an arbitrary pose with some garments, and as the result, how the person with the particular pose can be dressed realistically with a select set of garments chosen from a 3D-wardrobe. By proving the concept of this virtual try-on pipeline, it will be a crucial stepping stone to the future work in addressing a real-time rendering of a posed person to be dressed with an arbitrary set of clothing articles.

II. Introduction

The year 2020 was the year that quite literally plagued us with the COVID-19 pandemic, because of which people were asked to stay at home for an indefinite duration. Online shopping, even before the pandemic, already was a very popular option for modern shoppers, but the pandemic has accelerated the use cases and the necessity for it significantly. Shopping for clothes, however, remains as an experience that requires physical presence, because gauging how good a garment fits one is not an easy feat to be done virtually - the color scheme, garment shape can all be represented and quantified by measurements, but how would one *experience* the fit? In other words, failing the "fit" test influences one's like or dislike (hence the decision to buy or not) much more than any other features of the clothes.

Deep learning using computer vision has advanced significantly to be able to realistically augment the reality of the world that we live in. One of the very modern examples of this was the invention of Pokemon Go, where the virtual characters would sporadically *appear*, seemingly in the midst of the real physical world that our eyes see. Similar augmented reality approach can be combined with the cutting-edge computer vision algorithms to enable real-time rendering of the virtual try-ons of various garments. As aforementioned, this would transform the $1.46 trillion[9] fashion industry by converting the physical brick-and-mortar shop customers to shop confidently online for fashion apparels.

Many of the modern virtual try-on algorithms do not take into account the minute details of the body shapes, hence the wrinkles, stretches and other details of the garments are not captured. For the algorithms that do, there is not yet a seamless way to provide a simple input (i.e. photo) to get a simple output (i.e. extracted pose from the target to dress). Fortunately, there already is a wealth of research (mostly from Max Planck Institute for Intelligent Systems) that already solves many parts of extracting from the source [i.e. photo or video] then dressing the target. By utilizing some of the already available algorithms, this paper will attempt to prove the concept of extracting a pose, θ, from a 2D static image, then as the output, getting a dressed 3D rendering of a model of a human, with a garment (currently chosen arbitrarily from the Multi Garment Network (MGN)'s 3D wardrobe - see **Related Works**).

Fig. 1: Pose to model (My golf back-swing)

III. Related Works

- **ClothCap**[8]: Uses 4D scan sequence of single subject. 4D scan requires special devices that capture 4D time series data (of the 3D object), which make the data collection difficult.
- **Multi-Garment Network (MGN)**[1]: Predicts the body shape and clothing on top of the SMPL model. 712 different garments from the digital wardrobe can be used to dress any body shapes (compliant to SMPL model), and it contains the following garment types:
  - Pants
  - ShortPants (Shorts)
  - ShirtNoCoat (Shirt without coat, with which the occlusion of the garments complicate the MGN, hence not supported)
  - TShirtNoCoat (T-Shirt without coat)
  - LongCoat

  Each garments has been generated by calculating the displacement (equation 1), obtaining the unposed garment shape for new shape (β), and pose (θ) (equation 2), then eventually posing the unposed garment shape (equation 3).

$$D^g = G^g - I^g T(\beta, \theta, 0) + D^g \tag{1}$$

$$T^g(\beta, \theta, D^g) = I^g T(\beta, \theta, 0) + D^g \tag{2}$$

$$G(\beta, \theta, D^g) = W(T^g(\beta, \theta, D^g), J(\beta), \theta, W) \tag{3}$$

  where
  - β: new shape
  - θ: pose
  - D: per-vertex displacements
  - W: standard skinning
  - J: skeleton
  - T: base-mesh with n vertices
  - g: garment class
  - G^g: template mesh in T-pose
  - $I^g \in \mathbb{Z}^{m_g \times n}$: indicator matrix, with $I^g_{i,j} = 1$, if garment g vertex $i \in \{1...m_g\}$ is associated with body shape vertex $j \in \{1...n\}$
- **Skinned Multi-Person Linear model (SMPL)**[6]: "Realistic 3D model of the human body" based on CAESAR dataset. This model notably represents the soft-tissue deformations that adds realism to many unrealistic body deformations in other models. This is the basis model used in MGN.
- **SMPLify**[2]: Uses the CNN-based DeepCut[7] (which abstract away many SMPL joints into just 14 - see appendix I for the mapping information) to predict the 2D body joint locations then constructing a 3D posed model with a bottom-up process. The objective function (sum of the five error terms: $E(\beta, \theta)$ = equation 4) generatively optimizes the pose (θ) and shape (β), such that the projected joints of the 3D mesh model are close to the 2D joints estimated by the CNN, the result of which is surprisingly plausible.

$$\begin{aligned} &E_J(\beta, \theta; K, J_{est}) \\ &+ \lambda_\theta E_\theta(\theta) + \lambda_a E_a(\theta) + \\ &\lambda_{sp} E_{sp}(\theta; \beta) + \lambda_\beta E_\beta(\beta) \end{aligned} \tag{4}$$

  where
  - K: camera parameters
  - λ_θ, λ_a, λ_{sp}, λ_β: scalar weights

IV. Methods

In order to dress an arbitrarily posed body detected from a 2D input photo, we will need to use two algorithms. First, we will need to create a SMPL model of the detected pose from the 2D input photo using SMPLify[2]. Using 2D DeepCut joints predicted by the forward-pass SMPLify, it uses the generative CNN to create the

3D SMPL model. Using the resulting 3D SMPL model, `dress_SMPL.py` in MGN repository is used to dress the SMPL with the pose, θ, with a clothing articles chosen to evaluate the quality of the fit rendered. The analysis will be discussed in the later section of this paper.

A. *Using SMPLify to Create SMPL Model*

Although SMPLify[2] was designed to be used for the LSP dataset, I instead provided the image Figure 1 as the input. By running `fit_3d.py`, I was able to obtain the SMPL model in a `.pkl` format. The accuracy of the 2D DeepCut joints output will be cross-referenced with the results of the Openpose (output of which is also in 2D).

B. *Pre-process the Data for MGN*

Input images, as can be seen in Figure 1, are the cropped 720 x 720 2D static images taken with a standard camera (I have used the rear-view camera on *iPhone 12 Pro*). However, in order to use MGN for the forward pass (no back propagation, or hyperparameter optimizations), some pre-processes needed to be done.

- **Segmented images of a person**[5]: As shown in Figure 2, the semantically segmented output needed to be post-processed to denote white for skin, hair and other clothing articles, while keeping the shirt and the shorts to the suggested RGB mappings from the MGN repository:
  - Pants (65, 0, 65)
  - Short-Pants (0, 65, 65)
  - Shirt (145, 65, 0)
  - T-Shirt (145, 0, 65)
  - Coat (0, 145, 65)
  - Others (255,255,255)
- **2D joints** detected from Openpose (J_2d_x) [3][4][10][11]
- **vertexlabel** to generalize which SMPL vertex belongs to which garment. This can be generated from the garment template provided. For this experiment, we have labeled the SMPL vertices with the clothing articles T-Shirt and Short-Pants.

C. *Forward-pass MGN*

In the MGN repository, `dress_SMPL.py` has been used to dress the 3D mesh SMPLified model.

(a) Before processing (b) After processing

Fig. 2: Semantic segmentation process

(a) Dressing the 3D mesh with a t-shirt (b) Dressing the same 3D mesh with a short

Fig. 3: Dressed SMPLified 3D mesh model

To understand how the MGN was able to render the deformations from the detected pose, the corresponding rendering of the unposed (neutral, T-pose) model with the same garment is shared in the next section for the detailed analysis.

V. Experiment

The results of dressing the model with a t-shirt and shorts is shared in Figure 3. These clothing articles were chosen out of 25 t-shirts and of 19 shorts that are available in the MGN's 3D garment wardrobe. textifUnposed shows the garment rendered with the model in a T-pose, and *posed* is the based upon donning the garment on the posed SMPL 3D-mesh output.

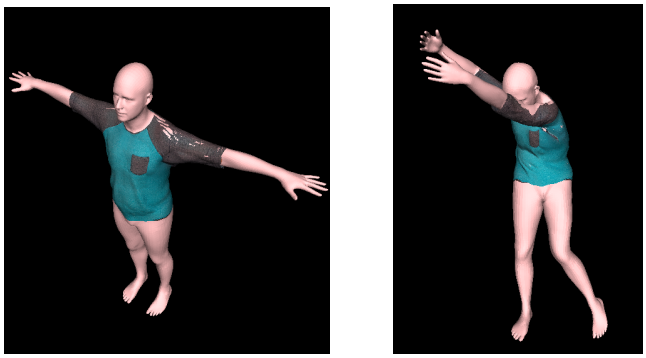
A. *T-shirt Fit Quality*

The unposed and posed rendering of the shirt is shown in Figure 4. In the unposed, there are *creases* at the joints (one in the upper arm, another in the lower), and the creases are there (with

unnatural appearance) because it is reconstructed from the various texture (i.e. `multi_tex.jpg`, `registered_tex.jpg`, `scan_tex.jpg`), along with the semantically segmented garments (i.e. `segmentation.png`) that identify which part of the body the garment belongs to. These inputs account for the successful reconstruction of the garment in 3D.

Overall, the posed-dressed rendering is fairly realistic - torso shows a swirl expected from a back-swing motion, and the left contracts to the body contour. Also, the right sleeve has a swirl pattern, which is expected as well from the motion. These two patterns match with the observed pattern in the input image in Figure 1. However, the the appearance is still nevertheless awkward - the left sleeve shows no swirl nor a slight sagging expected from a pose as such. Also, the left armpit area shows pathes (which seems like some remnant pattern of the sleeve color) and the top of the shirt shows lots of skin. This awkward deformation is likely resulting from the encoding error when registering the garment (since they were registered based on the neutral, T-pose), as the unposed-dressed shows holes near the shoulder.

(a) Unposed (T-pose) (b) Posed



(c) Unposed - dressed (d) Posed - dressed

Fig. 4: Dressing 3D mesh with t-shirt

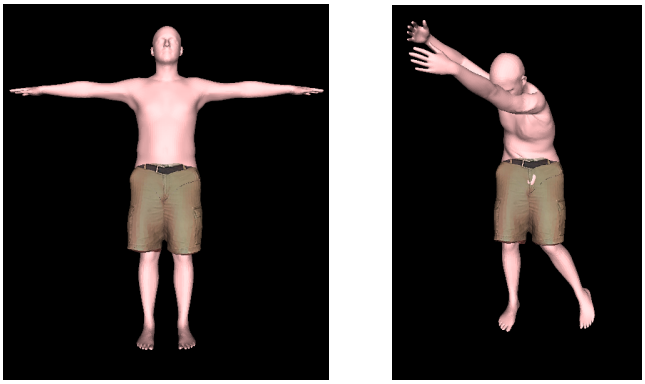
B. *Short Fit Quality*

The garment's renderings are rather disappointing for the shorts. As shown in Figure 5, the unposed and posed garments look almost identical, even though Figure 1's back-swing motion shows stark twist of the hip joints. This may be due to the SMPL model not properly capturing the twist of the hip joint, as evidenced by the posed-dressed in Figure 5 - even though the knee and the shins in motion were accurately extracted, the extracted hip joints seemingly did not show twist. There also seems to be an error in the wardrobe encoding in this rendering, because the belt (partially shown), which admittedly could be too complicated for the algorithm to distinguish, is shown rather as a part of the short. In addition, the bulging in front of the shorts shows fair amount of skin, which can potentially be attributed to the zipper-hiding overgarment being mistakenly encoded as a silhouette of a certain pose. In fact, it was suggested[1] to supply the silhouette of the frame for better results, which can be tested in a follow up study.

C. *Garment Encoding*

Since MGN uses semantic garment segmentation to RGB (images of which is denoted by $I = \{I_0, I_1, ..., I_F - 1\}$, where F is the number of images), it purposefully abstracts away the appearance information to reduce overfitting. However, as mentioned in the paper[1], it has the cost of disregarding useful shading signals, which is what is suspected to have happened on the T-shirt fitting. As seen in the first unposed garment, the shoulder area shows no holes (it shows *creases* on the left shoulder), but when it is donned on the model, the holes appear. The creases of the unposed garment, may have been abstracted away during the process of semantic segmentation, hence it resulted in the shoulder area to appear to have holes. Perhaps allowing certain hyperparameter that governs the level of model fitting (to balance between overfitting and underfitting) would help control these phenomenon.

(a) Unposed (T-pose) (b) Posed



(c) Unposed - dressed (d) Posed - dressed

Fig. 5: Dressing 3D mesh with short

D. *Distortion Reduction*

To reduce the distortion on the posed rendering, MGN attempts to preserve the L_2 distance between each vertices and the corresponding surface (i.e. vertex covering garment) before and after posing, according to the eq 5, where $d(x, S)$ is the L_2 distance between point x and surface S and the superscript ^{0} denotes that of the unposed model. This could explain how the distortion of the posed short may have happened - posing in a back-swing motion exposes more skin (hence more SMPL vertices) in the crotch that may have been occluded while unposed. Hence, the posed model now has some vertices that were previously occluded and consequently have no corresponding garment surface data. It seems likely that this can improve with the silhouette provided as input, since they provide gradation information of the garments (which in turn could provide useful leads to map some hidden, potentially un-mapped SMPL vertices).

$$E_{unpose} = \sum_{g} \sum_{v_k \in G^g} (d(v_k, S) - d(v_k^0, S^0))^2 \quad (5)$$

VI. Conclusion

The proof of concept of extracting the model from the 2D static photo, and donning a real set of garments were explored in this paper. The overall results were fairly realistic, granted that no further hyperparameter refinements (i.e. input silhouette, back-propgation MGN) have been done for this experiment. As aforementioned, many of the awkwardness in fit seems to be due to the way that the wardrobe was encoded, how the distortion reduction is being done, and also from the mechanism with which SMPLify captures the twist of subtle joints (i.e. hip joints). These, fortunately, seem only reflective of the shortcomings that can easily be enhanced, and not necessarily of the core algorithms used, hence some further algorithmic refinements might suffice. This experiment shows that automating the real-time rendering (MGN) of the garments on a arbitrary extracted human pose (SMPLify) is surprisingly robust and easily enhance-able, hence provides promising directions for the future researches.

VII. Future Works

Current work focuses on dressing the extracted SMPL/DeepCut model with the available clothing from the MGN 3D-wardrobe. Future work can use the wardrobe generating algorithm from the MGN to extract the clothing articles from the 2D static photo, instead of just the pose. This will potentially allow a new way to virtually try-on by using two inputs:

1) Extract the garments from the first photo,
2) Using the extracted garments, dress the second photo's pose

REFERENCES

[1] Bharat Lal Bhatnagar et al. “Multi-Garment Net: Learning to Dress 3D People from Images”. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE. Oct. 2019.

[2] Federica Bogo et al. “Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image”. In: *Computer Vision – ECCV 2016*. Lecture Notes in Computer Science. Springer International Publishing, Oct. 2016.

[3] Z. Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[4] Zhe Cao et al. “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *CVPR*. 2017.

[5] Ke Gong et al. “Instance-level Human Parsing via Part Grouping Network”. In: ECCV. 2018.

[6] Matthew Loper et al. “SMPL: A Skinned Multi-Person Linear Model”. In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.

[7] Leonid Pishchulin et al. *DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation*. 2016. arXiv: 1511.06645 [cs.CV].

[8] Gerard Pons-Moll et al. “ClothCap: Seamless 4D Clothing Capture and Retargeting”. In: *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 36.4 (2017). Two first authors contributed equally. URL: http://dx.doi.org/10.1145/3072959.3073711.

[9] M. Shahbandeh. *Global Apparel Market - Statistics Facts*. https://www.statista.com/forecasts/821415/value-of-the-global-apparel-market. Accessed: 2021-03-14.

[10] Tomas Simon et al. “Hand Keypoint Detection in Single Images using Multiview Bootstrapping”. In: *CVPR*. 2017.

[11] Shih-En Wei et al. “Convolutional pose machines”. In: *CVPR*. 2016.

APPENDIX

index	joint name	SMPL joint id
0	Right ankle	8
1	Right knee	5
2	Right hip	2
3	Left hip	1
4	Left knee	4
5	Left ankle	7
6	Right wrist	21
7	Right elbow	19
8	Right shoulder	17
9	Left shoulder	16
10	Left elbow	18
11	Left wrist	20
12	Neck	-
13	Head top	vertex 411

TABLE I: DeepCut joint to SMPL joint mapping