Body Capture and Marker-based Garment Reconstruction

Zhengping Zhou, Daniel Do, Alice Zhao, Jenny Jin, Ronald Fedkiw

UGVR Program, Stanford University

Abstract. In this work, given videos of a moving person from multiple calibrated RGB cameras, we present a marker-based method to get a 3D animation for both the person and the garment. Previous works on RGB videos either only extract the person / garment, or mix them together as a single surface. In contrast, our system simultaneously captures the body and the garment as 2 separate surfaces. Our approach starts from digitizing the garment by triangulating the boundary of scanned pieces. Afterwards, we track the markers across frames to get their 3D locations as a set of linear sparse constraints. We then optimize over an animatable body template, SMPL, to obtain a body model that is accurate both in pose and shape. Finally, we adopt the level set approach to virtually "wear" the garment on the body.

Keywords: Body Model, Garment Reconstruction, Optimization

1 Introduction

Our goal is to generate a 3D model of *a person wearing a garment*, from multi-view RGB videos. Such an accurate model would be useful for many special effect applications, e.g. garment re-targeting or VR transfers. However, existing methods, though already achieving great progress, are still far from satisfactory.

One mainstream approach for garment reconstruction stems from physical simulation. In the computer graphics community, researchers have achieved quite nice simulation of many different types of cloth using a variety of interesting techniques; however, their ability to match real cloth of specific material, especially with highly detailed wrinkling, hysteresis, and other real-world so-called imperfections is rather limited. For example, (Bhat et al., 2003) first estimates the simulation parameters from a small patch, then apply them on the full garment. However, the physical models are always imperfect, and even if the simulated result looks plausible at the first glance, they still miss many verisimilitude details.

Our method lies in another important branch, i.e. *garment capture*. Instead of establishing a physical model and computing the surface according to the stress analysis, we directly capture the high-level geometry of garment, by deforming the surface to fit a set of sparse markers. (Bradley et al., 2008) captures a smooth surface for the garment by setting up a stereo of 16 cameras, yet they do not generate a model for the person inside. There are also some works focused on

single-view use cases, e.g. (Danek et al., 2017) adopts deep learning techniques for dynamic garment capture in a single image, which is a different scenario from ours. Our approach leverages the multi-view information, and outputs a model for the person inside as well. We enforce the collision constraint between the garment and the body using a level set approach, and hence also avoid interpenetration for some edge cases.

Another important aspect for our approach is the need for a high-fidelity animatable body model, since a 3D model for the person is also a desired output. We adopt the SMPL (Loper et al., 2015) body model as an parameterized animatable template, which is basically a differentiable surface w.r.t. body shape and pose. We first take the person's height and girth measurements at different body positions, then use joint detectors to estimate the 3D joint locations. We later run conjugate-gradient optimizations over the SMPL model w.r.t. shape and pose, to fit the ground truth measurements and detected joints. There are many similar works aiming at extracting a body model from a person's photo, yet to the best of our knowledge, none of them leverages the body measurement information, and most of them tend to generate fatter bodies due to errors in the silhouette introduced by the garment. Qualitative and quantitative analysis demonstrate that our body capture system is able to generate a body model with accurate pose and shape.

Given the digitized garment, the 3D marker positions, and an accurate body model, we finally run a bounded optimization problem to virtually "wear" the garment on the avatar. This is achieved by setting up virtual springs to ensure smoothness, as well as following the sparse constraints provided by the markers. We also build a level set for the person to discourage the garment from penetrating the body. Our method achieves decent looking clothed 3D person as a final output.

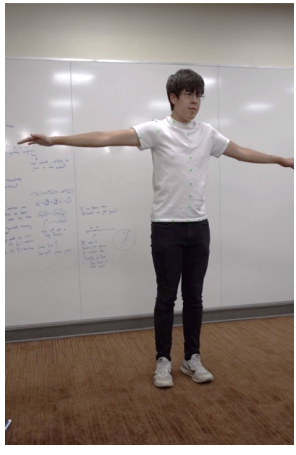

Fig. 1. Input

Fig. 2. Output

2 Related Work

Garment Capture This class of methods only capture the garment as an output. (Pritchard et al., 2003) uses SIFT features to establish the correspondence between a flat garment and a worn garment. It requires the garment to have a non-repeating and unique pattern, so that the SIFT features could be as robust as possible. (Pullen et al., 2005) leverages color codes printed on garment as spatial hints, and successfully capture garments with decent looking outcomes. (Bradley et al., 2008) proposes a marker-free method to capture the garment surface, by doing interpolation using a 16-camera stereo. (Popa et al., 2009) further adds wrinkles and high-frequency details to the work of (Bradley et al., 2008), achieving a more vivid result.

Although those methods are able to capture details that are difficult to model by a physical simulation, they have 2 main drawbacks: First they do not extract a body model from the video, second most of them require an unfeasible setup, such as special patterns to be printed on garment or too many devices in a carefully-organized studio. Our experiments, in contrast, are easy to setup, and only need to put a small amount of markers temporarily on the garment.

Body Capture There are extensive works aiming at extracting a body model without clothing from a person's photo or video. Most recent works are based on SMPL (Loper et al., 2015), the Skinned Multi-Person Linear Model, which is a rigged and skinned differentiable body model parameterized by body shape, body pose, and a global translation. It enables researchers to directly fit the shape or pose via numerical optimizations. (Bogo et al., 2016) introduces SMPLIFY, a system that takes in the focal length and extracts a naked body model from a single image. It first uses a 2D joint detector to get the joint locations, then also use a shape prior to deal with depth ambiguity. It also avoids penetration by approximating each body part as a cylindrical capsule. The entire model is optimized using a dogleg trust region method, which is followed by many later works. (Kanazawa et al., 2018) builds an end-to-end network for a similar single-view use case.

This category of approaches, though straight-forward, do not work well in our use case. The first issue is they do not leverage the body measurement information, which is easy to obtain in a special effect application. They tend to generate fatter and shorter body models, due to cloth offsets and perspective distortions. The second issue is the multi-view inconsistency. Due to the inherent ambiguity in depth for single-view methods, they tend to generate different models in different views. (Pavlakos et al., 2017) builds a 3D multi-view probabilistic optimizer that takes in the 2D joint probability distributions, then predicts the 3D joint locations by taking the expectation of the 3D probability distribution. (Huang et al., 2017) proposes a similar multi-view body reconstruction system, yet they produce the entire body surface rather than only joint locations, making it harder to integrate and customize. We finally adopt the approach in (Pavlakos et al., 2017) to generate 3D joint locations as an intermediate output, then cus-

tomize the SMPL model according to our body measurements and the 3D joint locations.

Person and Garment Capture Some works capture the person and the garment at the same time, and most of them generate a single mesh fusing the person and the garment together. Some typical examples include (Gall et al., 2009), (DeAgiar et al., 2010), (Allain et al., 2014) and (Neophytou et al., 2014). This kind of algorithms can be problematic when it comes to the rendering of adjacent regions of garment and flesh, where textures can mess up due to the ambiguity of mesh fusion.

A recent work, (Pons-Moll et al., 2017) is able to reconstruct separate meshes for the person and the garment, which takes in high-quality data from 4D scans, making it to be more demanding for devices and resources. Whereas, our method only requires 3 RGB consumer cameras, and also outputs decent and separated meshes for the person and the garment, respectively.

3 Method Description

Given a garment and multi-view videos of a person wearing it, our system is separated into several stages:

1. **Garment Digitizing:** Digitize the garment into a 3D flat mesh.
2. **Marker Tracking:** Track the markers and obtain their 3D locations.
3. **Body Capture:** Reconstruct a body model with accurate shape and pose.
4. **Garment Reconstruction**: Virtually wear the garment on the body.

3.1 Garment Digitizing

Given a garment from the physical world, there are 2 steps for generating a corresponding 3D mesh:

2D Mesh Generation In this step, we generate a 2D design pattern as an intermediate representation. This can be done either in Marvelous Designer (a non-free design software, referred to as "MD" below), or by triangulating the boundary of scanned pieces.

Marvelous Designer (MD) Take a photo of the garment right above the garment, import that photo into MD, then follow the boundary to generate the 2D design mesh.

Scanner Cut the garment into pieces small enough to be covered by the scanner, then manually merge them together in PhotoShop (also somehow smooth the boundary). Use the magic stick tool to get a black-white image as background subtraction. For symmetric parts (e.g. legs for the jeans), this is only done once and then duplicated and mirrored.

Given the background subtraction of a scanned piece, we first extract a dense set of points on the contour, then uniformly sample points on each edge specified by the user. Finally a Delaunay triangulation is used to obtain a triangular mesh.

3D Garment Stitching In this step, we stitch the generated 2D design pattern pieces together. This can be done either in Marvelous Designer, or using our flat stitching script.

Marvelous Designer (MD) After generating the 2D pattern in MD, select the seams to be stitched together, then run the simulation. In the material panel, turn the bending/warping/stretching all to 0, and keep all other settings as default. You should be able to get a decent looking and roughly flat 3D mesh afterwards (with reasonable deformation on curved parts).

Flat Stitching The flat stitching tool is limited to flat garments (e.g. T-shirt). Any non-coplanar garments with non-negligible deformation along the normal (e.g. middle seam of the jeans) wont be properly handled.

The tool takes in manual annotations on the pieces, front/back, and seams. It comes with a blender GUI to make the annotation process easier. It stitches each seam one by one: For each seam, it first finds the optimal 2D rigid body transform R, t between the 2 related pieces (also forbids reflection if necessary):

$$S = (P - \bar{p})^T \cdot (Q - \bar{q}) \tag{1}$$
$$U, s, V = svd(S) \tag{2}$$
$$R = V \cdot U \tag{3}$$
$$t = q - R \cdot \bar{p} \tag{4}$$

Then merges the vertices along the seam at the center.

3.2 Marker Tracking

To locate and translate the markers into a group of sparse constraints on the garment, there are 3 major steps to do:

Barycentric Coordinates First we need to determine the barycentric embedding of each marker on the generated garment mesh:

$$v = \alpha v_1 + \beta v_2 + \gamma v_3$$

where v is the location of the marker, and v_1, v_2, v_3 are the vertices of the containing triangle.

Marker Tracking

Old Tracking Method Previously, we first detect blobs in each input frame, then use a simple heuristics to track them across frames. This method is bottlenecked by the accuracy of blob detection (low precision or recall in blurry frames or poor lighting conditions). Afterwards we manually label each detected blob with the corresponding marker ID, and usually for a 6 secs video and 12-15 visible markers we need to manually label 40-50 times, because this method is super fragile to temporary occlusion or detection failure.

New Tracking Method Now we discard the old detection + tracking framework, and fully tackle this as a multi-object tracking problem. The new method first asks the user to draw a bounding-box for each marker in the first frame, then use the CSRT tracker (superior on our videos, as compared to any other tracker in OpenCV) to track each bounding-box. The marker location is detected as the maximum magenta value (channel a in LAB color space) in the bounding-box. This method is very robust to short occlusions, and we only need to label roughly the same number of times as the total number of markers. The drawback of this method is that it ignores new marker entrance.

Marker Amending However, tracking failure still occurs in some tough videos. Hence, we built an amending tool. It currently supports modifying the id of a blob starting from a certain frame, or deleting a blob in one frame. This resolves the 2 most frequent failure cases. In our new method, on a 10 secs, 30 Fps video with 12-15 visible markers, the required times of amending is usually no more than 5 (and for most times 0).

Marker Triangulation Given 2 stereo parameters, marker positions in 3 views, and mappings from blob ids to marker ids, we do a multi-view triangulation to get the 3D locations of markers in each frame. We do the triangulation by using camera 1 as the reference coordinate system, and triangulate in stereo pair 1-2, 1-3 separately, then take the average.

3.3 Body Capture

In this step, we get a body model with accurate pose and shape. We do this by running conjugate-gradient optimizatoins over the parameterized SMPL body model $S(\beta, \theta, t)$, where $\beta \in R^{10}$ represents the body shape, $\theta \in R^{72}$ stands for the joint rotations, and $t \in R^3$ is for the global rigid translation. There are 2 major steps to do:

Body Shape We tried 2 different methods for generating a high-fidelity body shape. The first one takes in body measurements as constraints, and the second one projects the mesh vertices from an existing model to the SMPL body model to compute the 10 PCA principle components. We found the first one to be more accurate in practice, yet list them here together for completeness.

Measurement Method Our goal is to fit the SMPL body model to the girth measurements and height of the person. So as a preprocessing step, we first measure a predefined set of girths for the person in T-pose, and then represent them using the barycentric xoordinates of on-loop points, which are computed by cutting a plane across a canonical SMPL model in T-pose. We then run a CG optimization over the girth measurements and the height, w.r.t. β . We also add a L2 normalization on β so we will not be over-fitting and get some weird looking results.

Projection Method We also tried to project an existing body model (non-animatable) onto an SMPL body model. This is done by first aligning the SMPL body model to the same pose as the existing model, then project the SMPL vertices onto the nearest neighbor on the target body. The shape parameter β is computed by taking the dot products between the 10 PCA basis and the difference between the SMPL canonical template and the projected mesh.

Body Pose In order to obtain the body pose, we first run a 2D joint detector to get heatmaps, then run a 3D multi-view optimizer to probabilistically do the triangulation according to the heatmaps. Afterwards, we run a CG optimization over SMPLś θ, t (i.e. pose and translation) to get a final body model. We finally attract the body to the markers to get a better fit.

2D Joint Detection We use the Hourglass Network by (Newell et al., 2016), which is a state-of-the-art model for 2D joint detection.

3D Joint Optimization We use the method proposed by (Pavlakos et al., 2017), which runs a multi-view probabilistic optimization for the optimal 3D joint locations.

Joint Fitting We run a conjugate-gradient optimization over the joint positions, namely

$$E_{joints} = \sum_{j \in joints} w_j \| J \cdot S_\beta(\theta, t) - j \|^2$$

where w_j are customized joint weights, J is the SMPL joint regressor, and S_β is the SMPL model function $S(\theta, \beta, t)$ with fixed β.

Marker Attraction We run a conjugate-gradient optimization to attract the body to the markers, also discourages penetration by adding a huge penalty, namely

$$E_{markers} = \sum_{m \in markers} \| NearestNeighbor(m, S) - m \|^2 + E_{penetration}$$

$$E_{penetration} = \sum_{m \in markers} \| \min(0, \mathbf{n} \cdot (m - f)) \|^2$$

Where $\mathbf{n}, f$ are the normal vector and the centroid of the nearest triangle on SMPL body model to marker m. This step is necessary because the joint detector isnt perfect and there can be some error in translation.

3.4 Garment Reconstruction

Given the 3D garment mesh, marker locations, and the body model, we run a L-BFGS-B optimization over the garment. This problem must be bounded because we cant exceed the boundary of the level set for the body model.

Initialization We first compute the global optimal rigid body transformation from the markers on garment to markers in world, then transform the garment accordingly to make it roughly aligned to the correct position. We then align the front and back pieces to be roughly tangent with the chest and back separately. Note that this only works for not too extreme poses such as A-pose or T-pose.

There are 4 energy terms in total. Note that the analytical jacobian (or at least a pre-computed numerical one) for each energy term must be explicitly provided, or the optimization will be unbearably slow.

Spring Energy We prevent stretching or deformation for each edge. We also add bending springs for each pair of triangle (except for those crossing front and back).

The energy term and the jacobian are as follows:

$$
\begin{aligned}
E_{spring} &= \sum_{e} \left(\frac{\|\mathbf{p}_i' - \mathbf{p}_j'\| - \|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j\|}{\|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j\|}\right)^2 \\
\frac{\partial E_{spring}}{\partial \mathbf{p}_i} &= \sum_{\mathbf{p}_j \in neighbor(p_i)} \frac{dE_1}{dr_{ij}} \cdot \frac{\partial r_{ij}}{\partial \mathbf{p}_i} \\
&= \sum_{\mathbf{p}_j \in neighbor(p_i)} \frac{1}{\bar{r}_{ij}^2}(2r_{ij} - 2\bar{r}_{ij}) \cdot \frac{1}{r_{ij}}(\mathbf{p}_i - \mathbf{p}_j) \\
&= 2 \sum_{\mathbf{p}_j \in neighbor(p_i)} \frac{1}{\bar{r}_{ij}^2}(1 - \frac{\bar{r}_{ij}}{r_{ij}})(\mathbf{p}_i - \mathbf{p}_j)
\end{aligned}
$$

Constraint Energy We use the markers as a set of linear sparse constraints over the garments. We construct a sparse matrix correspondingly to formulate the constraint problem.

The energy term and the jacobian are as follows:

$$
\begin{aligned}
E_{constraint} &= \|Ax' - y\|_2^2 \\
\frac{\partial E_{constraint}}{\partial \mathbf{x}} &= 2\mathbf{x}^T(A^T A) - 2\mathbf{y}^T A
\end{aligned}
$$

Penetration Energy We adopt a level-set and pre-compute the numerical derivative for each grid point. The signed distance function and jacobian are linearly interpolated at run time.

$$
E_{penetration} = \varphi(person)^2
$$

4 Experiments

4.1 Mesh Generation

We compare different methods for generating a digitized version of garment in this subsection.

It seems that MD is better at the garment generation, rather than our own heuristics. As for the 2D mesh generation phase, it generates a finer and nicer looking triangular mesh, while ours look more like a disturbed square grid; As for the 3D garment stitching process, it handles deformations properly, and their GUI is also much more user-friendly than ours. Although scanned pieces may be more faithful to the original shape, the mesh diff shows that it actually generates similar outcome, as compared to a quick 2D mesh generation in MD.

Table 1 shows the quantitative comparison for a jeans generated using different methods. The mesh diff and other relevant measurements all demonstrates that they actually have very small differences. Figure 3 shows a quantitative comparison. There seems to be a visible difference across the middle seam, yet this part wont́ be greatly affecting the simulation, and is still within a bearable range.

Table 1. Garment Mesh Measurement Comparison

	In-Leg	**Leg-Open**	**Out-Leg**	**Belt**	**Waist**	**Inseam**	**Back-Waist**	**Back-Inseam**
GT	75	19.5	96	4	42	25	41	30
MD	75	19.5	96	4	42	25	41	30
Scanned	75	20	100		38	29	38	32.7
Mesh Diff						5mm		

Fig. 3. Garment Digitizing Comparison

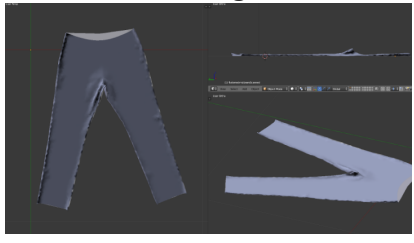

Fig. 4. Scanned Pieces

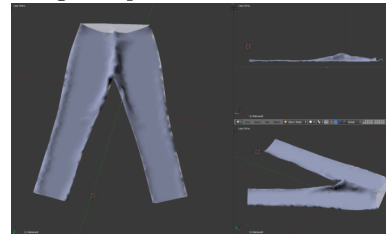

Fig. 5. Marvelous Designer Pieces

In summary, as long as you 1) Can get a MD 30d free trial or buy a license; 2) do not require the 3D garment mesh to be perfectly flat (so its easier to manipulate and hack later), personally I would recommend using MD for digitizing a garment.

4.2 Body Shape

We found the first method, i.e. directly optimizing over the body measurements, to be more accurate. We compare different methods both qualitatively and quantitatively.

Table 2 shows the quantitative comparison in terms of body measurements. The first method (labeled as "M") tends to be consistently achieving lower errors. Figure 6 shows a qualitative analysis case. Note the difference under the armpit. By directly optimizing over the body measurements, we are able to achieve a more accurate body model in terms of body shape.

Table 2. Body Measurement Comparison

Measurement	Value(M)	Value(P)	Ground Truth	Percent(M)	Percent(P)
Height	1.7898	1.7835	1.8	-0.57%	-0.93%
Upper Arm	0.4567	0.4818	0.36	21.17%	25.28%
Upper Chest	1.0206	1.0943	0.89	12.79%	18.67%
Middle Shin	0.3501	0.359	0.39	-11.41%	-8.63%
Upper Thigh	0.5879	0.6121	0.53	9.85%	13.41%
Wrist	0.1618	0.1647	0.148	8.54%	10.12%
Lower Neck	0.4101	0.4257	0.39	4.89%	8.38%
Upper Shin	0.335	0.3451	0.35	-4.46%	-1.43%
Bust	0.9159	0.9971	0.876	4.36%	12.15%
Elbow	0.2462	0.2632	0.24	2.52%	8.82%
Lower Thigh	0.383	0.3985	0.39	-1.82%	2.13%
Under Bust	0.8616	0.9372	0.846	1.81%	9.74%
Hips	0.9169	0.9546	0.93	-1.42%	2.58%
Waist	0.7979	0.8701	0.788	1.24%	10.12%
Lower Shin	0.2228	0.2245	0.223	-0.09%	0.69%

Fig. 6. Body Shape Comparison (Note the armpits are different)

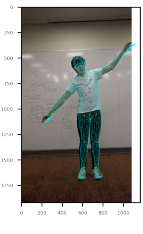

Fig. 7. Measurement Method

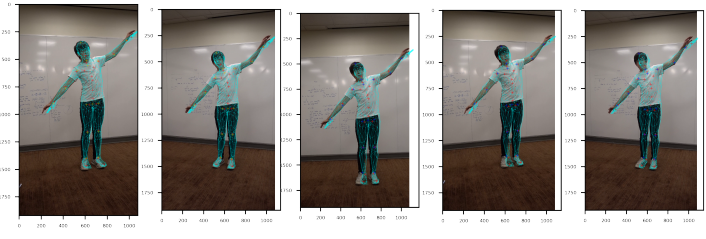

Fig. 8. Projection Method

4.3 Garment Reconstruction

We are finally successful to virtually "wear" the garment onto the reconstructed body model by running the optimization described in the above section. Figure 9 shows a running demo:

5 Conclusions

In this project, we propose a new method for generating a 3D animation for both the person and the garment from multi-view RGB videos. Our intermediate results are discussed and compared both quantitatively and qualitatively, and We achieve decent looking results in the end.

References

1. As-Rigid-As-Possible Surface Modeling (2007 Sorkine et al., EUROGRAPHICS)
2. Cloth Motion Capture (2003 Pritchard et al., EUROGRAPHICS)
3. Cloth Parameters and Motion Capture (2001 Pritchard et al., EUROGRAPHICS)
4. Garment Motion Capture Using Color-Coded Patterns (2005 Pullen et al.)
5. A Survey of Computer Vision-Based Human Motion Capture (2001 Thomas et al.)
6. Estimating Cloth Simulation Parameters from Video (SIGGRAPH 2003, Bhat et al.)
7. Research problems in clothing simulation (2005, Choi et al.)
8. Markerless Garment Capture (SIGGRAPH 2008, Bradley et al.)
9. Wrinkling Captured Garments Using Space-Time Data-Driven Deformation (EUROGRAPHICS 2009, Popa et al.)
10. SMPL: A Skinned Multi-Person Linear Model (SIGGRAPH Asia 2015, Loper et al.)
11. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image (ECCV 2016, Bogo et al.)
12. End-to-end Recovery of Human Shape and Pose (CVPR 2018, Kanazawa et al.)
13. Harvesting Multiple Views for Marker-Less 3D Human Pose Annotations (CVPR 2017, Pavlakos et al.)
14. Towards Accurate Marker-Less Human Shape and Pose Estimation over Time (3DV 2017, Huang et al.)

Fig. 9. Final Result