Internship Report: 3D Human Reconstruction Methods using a Latent Representation of Human Form

Joel Huang

Singapore University of Technology and Design
NCS Product R&D, Singapore

31 August, 2018

Abstract

Recovering 3D spatial information about humans from a single photograph or video frame can be assisted by posessing prior knowledge of human form. In human vision, we have learnt, by experience and from birth, a mental model of what a person is supposed to look like. Similarly, we can learn a latent representation of human form that machines can understand. We do not understand our own form in three dimensional space - using a parametric 3D body model, we can better represent the human form in its principal components. Using this model, we can directly estimate body model parameters from photographs, reconstructing human form that is both statistically representative of the human population, and accurate to the context of the image.

1 Introduction

Once we teach machines to see us as we do ourselves, we unlock a huge realm of possibilities, a future of which post-screen interfaces, vision-based analytics, and hyper-realistic simulation are a part of. Traditionally, much of the work in this field has focused on the problem of 2D **pose estimation**, defined as the localization of human joints. While this abstraction of human form is intuitive and has recently produced stunning results [4], it fails to retain information about the human form in three-dimensional space.

There has been a series of works focused on recovering the human form from photographs using a latent representation of the human body. Parametric deformable surface models such as the Skinned Multi-Person Linear (SMPL) model [10], a learned 3D model of human shape from thousands of body scans of different people, emulate reality where changes in human pose are coupled with body surface deformations. These models have been used in human **correspondence estimation**, in which a mapping between 2D pixels and the 3D model's surface is learned [5, 6].

It is possible that the human visual cortex also works with latent, learned representations of objects and forms that weight the estimations and decisions we make in our perception of form, depth and space. The authors of SMPLify [3] work along those lines, attempting the fitting of the 3D SMPL model to raw 2D keypoint detections. To do this, the parameters of the model which capture pose and shape must be optimized such that their combination results in the least error when reprojected to the plane of the original image. They introduce constraints in their iterative optimization approach, by favouring possible poses, penalizing impossible poses, and penalizing information from occluded body parts. Although iterative optimization has the least error among current methods, fitting for a single image is on the order of tens of seconds.

Once again, deep learning approaches have emerged to tackle the problem of efficiently estimating these parameters. Kanazawa *et al.* [8] propose an end-to-end encoder-regressor-discriminator framework which directly converts images into pose and shape estimates, while other approaches further decompose the problem into several smaller tasks, enabling training with samples that are easily obtained, such as 2D joints or shape segments. Omran *et al.* [12] decompose the problem this way, generating body part segmentation maps and using them to directly train a convolutional neural network, which then predicts the pose and shape parameters. Pavlakos *et al.* [13] separate the problem into modular tasks, using a stacked hourglass network to produce joint locations and silhouette masks, which

are then processed separately by two networks to obtain the pose and shape parameters respectively.

2 Modelling Human Form

2.1 Principal Components

When trying to capture spatial information, we often fall back to a three dimensional worldview, force-fitting a Cartesian parametrization for ease of understanding: we describe objects as taller, wider, and closer than one another; objects rotate about a defined global axis; and scale according to three familiar principal components, height, width and depth. In fact, it is sufficient to fully describe the spatial form of all objects, and is the most efficient way to describe regular shapes like cuboids and planes. But it does not mean anything useful when we describe human form using a set of three-dimensional vectors. We want to describe human form based on its principal components; features along the lines of stature, body mass, bone structure, mass distribution, fatness, skinniness, and others we subconsciously observe but do not have the words to describe.

2.2 Statistical Model

The Skinned Multi-Person Linear (SMPL) model is an example of describing the human form in Euclidian space, but not a Cartesian representation. The model is parameterized by 72 pose and 10 shape variables. As SMPL captures correlations in human shape across the population, we are able to robustly fit it to very little data.

2.2.1 Pose and shape

Pose is modelled as the set of axis-angle representations of 23 pre-defined body parts in the skeletal rig, plus the root orientation. $\vec{\theta}$, the pose parameters, are defined as:

$$\vec{\theta} = [\vec{\omega}_0^T, \ldots, \vec{\omega}_k^T]^T \tag{1}$$

where $\vec{\omega}_k \in \mathbb{R}^3$ is the rotation of each part k in Euler angles. The total size of the pose vector is $|\vec{\theta}| = (23 + 1) \times 3 = 72$. By modelling pose as joint rotations, the model can easily be used in existing rigging and rendering software.

Shape is modelled as weighted changes in orthonormal principal components, which define the shape space for the particular model. The principal components, which have been learnt from registered training meshes using Principal Component Analysis (PCA), are reminiscent of concepts of human form, like stature, or fatness. Each shape coefficient, β, weights a particular element in the shape displacement matrix, allowing us to modify, for example, the stature of a model simply by changing the appropriate β in $\vec{\beta}$:

$$\vec{\beta} = [\beta_0, \ldots, \beta_{|\vec{\beta}|}]^T \tag{2}$$

2.2.2 Blending pose with shape

Vertex deviations from the rest template due to the pose are accounted for by introducing a linear effect of **pose blend shapes** [10]. With 23 joints, the function that maps pose, $\vec{\theta}$, to part-relative rotation matrices, is $R(\vec{\theta})$, a vector of length $23 \times 9 = 207$. The final effect of pose on vertices is hence:

$$B_P(\vec{\theta}; \mathcal{P}) = \sum_{n=1}^{207} (R_n(\vec{\theta}) - R_n(\vec{\theta^*})) \cdot P_n, \tag{3}$$

where the difference between the detected pose and the rest pose weights each P_n in $\mathcal{P}$ linearly, and $\mathcal{P} = [P_1, \ldots, P_{207}]$ is a learned matrix of vertex deformations due to pose.

The vertex deformations based on the shape parameters are referred to as **shape blend shapes** [10], the observable effects of which are described in the previous section. Each shape coefficient β transforms vertices by:

$$B_S(\vec{\beta}; \mathcal{S}) = \sum_{n=1}^{|\vec{\beta}|} (\beta_n) \cdot S_n, \tag{4}$$

where $\mathcal{S} = [S_1, \ldots, S_{|\vec{\beta}|}]$ is the learned shape displacement matrix of vertex deformations due to shape.

The total deviation of vertices due to pose and shape blend shapes can simply be added to the rest pose $\bar{T}$, to recover the final vertex positions after accounting for shape and pose:

$$T_P(\vec{\beta}, \vec{\theta}) = \bar{T} + B_S(\vec{\beta}) + B_P(\vec{\theta}) \tag{5}$$

2.2.3 Rendering vertices

If we were to simply apply the pose to the model after adding pose and shape blend shapes, we would encounter errors, such as candy-wrapper artifacts, if joints are rotated at more extreme angles. Without blend skinning, which prevents errors in object surfaces when skeletal rigs are twisted or bent to unusual

angles, and the learned blend weight matrix $\mathcal{W}$, which tells us how much the rotation of part k affects each vertex, the rendered vertex mesh does not produce good enough results.

A blend skinning function returns the smoothed locations of vertices around estimated joint centers. Here, we introduce Linear Blend Skinning (LBS), but other methods such as Dual-Quaternion Blend Skinning (DQBS) exhibit different behaviors with regards to surface folding and twisting. The smoothed vertices are computed by the standard LBS function $W(T, J, \vec{\theta}, \mathcal{W})$.

2.2.4 Estimating joints

Joints are defined as a function of the body shape, with a need for high accuracy since they are to be used in the skinning equation. A joint regressor matrix, $\mathcal{J}$, is learnt, by predicting training joints from training bodies. The result is the function $J(\vec{\beta})$, which predicts joints from shape parameters β.

2.2.5 Final model

The final SMPL model M, is hence given by:

$$M(\vec{\beta}, \vec{\theta}; \Phi) = W(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}) \qquad (6)$$

where Φ represents learned parameters $\bar{T}, J, \mathcal{W}, \mathcal{S}, \mathcal{P}$.

2.3 Modelling non-body deformations

In their recent work on 3D human model recovery from video [2], Alldieck *et. al* propose the recovery of more than just the base human surface, including shape deformations from objects like hair, clothing, and headgear. They augment the SMPL model, introducing a set of auxiliary variables, D, as offsets from the SMPL template:

$$T(\vec{\beta}, \vec{\theta}, D) = \bar{T} + B_S(\vec{\beta}) + B_P(\vec{\theta}) + D \qquad (7)$$

3 Reconstruction Pipeline

We look at possible methods of recovering the inputs needed to reconstruct parametric models like the SMPL model. In particular, we discuss the subsystems proposed in [2, 3, 8, 12, 13] from a design perspective. Two camps emerge - methods that use different modules to execute specific tasks like joint regression or segmentation, and methods that directly recover parameters in an end-to-end fashion.

3.1 Modular Design

Modularity might be a possible guiding principle to approach the design of a real-time reconstruction pipeline, as it allows for modules to be improved separately, while allowing the use of the intermediate outputs to augment or process other information. For 3D human reconstruction in particular, the intermediate state is often some set of 2D joint confidence maps, body part segmentation masks, or silhouette masks.

In an end-to-end learning task where the sole objective is to learn the function mapping from an image to a 3D representation, the network might not be able to realise the significance of this set of outputs. Intermediate results like joints, confidences, and masks are often useful further down the pipeline. For example, Alldieck *et al.* [2] make impressive use of 2D silhouette segmentations, associating every point on the silhouette edge with a point on the SMPL model, then computing the inverse pose of this outline, finally obtaining a visual hull of the input human in T-pose, which massively simplifies the shape fitting of a T-posed SMPL model to this hull.

More obvious modular pipelines have been used with success, with division of labour among networks along the pipeline, each tackling a specific subproblem. However, one of the downsides to modular design is the compounding of error. For example, if a pipeline consists of a 2D joint detector and a 3D regressor module, a possible failure in joint detection can completely change the behaviour of the 3D regressor module. For this reason, it might be advisable, though counterintuitive, to further separate the problem into simpler tasks where we can have more control over the stages of processing.

3.2 Modular Subsystems

In this section, we discuss specific subsystems in the 3D human reconstruction ecosystem. They typically tackle specific tasks in the reconstruction problem and produce intermediate representations in both 2D and 3D.

3.2.1 2D feature extractors

From a monocular input image, classical low-level image processing objectives can be met by convolutional layers. They are typically included at the start of 2D pose estimation networks [4, 12, 15] to provide feature extraction capabilities. Fast, direct bounding box predictors like YOLOv3 [14] can be used to crop

regions of interest at tens of frames per second or more.

3.2.2 Pose estimators

Pose estimators might output features like hard joint keypoints and segmentation masks, softer or more probabilistic predictions in heatmaps, or totally different representations of pose like part affinity fields [4]. Part affinity fields are a set of 2D vector fields which encode the degree of association between parts: for each pixel in a body part segment (e.g. a forearm), we generate a 2D vector pointing from the origin of the limb, to the end of the limb. With part affinity fields, we are able to score and match candidate limbs with global context, due to the wide receptive field of the network. This results in high quality predictions, and enables a tree optimization algorithm that massively reduces inference time.

Stacked hourglass networks [11] were used to generate part heatmaps and silhouette masks [13]. Stacked hourglass networks progressively pool and upsample, capturing features at multiple scales and bringing them together to output pixel-wise predictions.

3.2.3 Regressors

Pose estimators output part segmentations and/or joint coordinates, which themselves can be used as ground truths in order to fit the body model. This has been done using the SMPL model's joint regressor function (see 2.2.4). In SMPLify [3], iterative optimization is used, with a complex objective function that imposes constraints on possible poses, while doing its job of minimizing error between the 2D joint predictions from the previous stage, and the joints computed from the joint regressor function. The five error terms to be minimized in the objective function are as follows:

$$E(\beta, \theta) = E_{joint} + E_{ext} + E_{pen} + E_{\theta} + E_{\beta} \qquad (8)$$

where:

1. E_{joint} is the weighted 2D distance between input estimated joints and $J(\vec{\beta})$ from SMPL,
2. E_{ext} is an exponential error term heavily penalizing hyperextended elbows and knees,
3. E_{pen} is an error term penalizing the intersections of 3D shapes corresponding to body parts,
4. E_{θ} is a pose prior: a gaussian mixture model of 1 million poses from real subjects, and
5. E_{β} is a shape prior: a matrix of coefficients β estimated via PCA from training set shapes.

Optimization is carried out, with the camera translation estimated. If auxiliary variables are added to the objective function as in (7), there must be an additional way to calculate their error. It is likely that this requires segmentation masks to compute the error between the silhouette clothing mask and the clothing deformation's 2D projection.

Lassner *et. al* [9] train 91 keypoint detectors, using random regression forests to directly regress SMPL model parameters from 2D keypoints. They manage to reduce runtime at the cost of accuracy.

3.2.4 Direct estimators

Alternatively, fully-connected and convolutional approaches to estimate θ and β have also been successful. In their recent paper [13], Pavlakos *et al.* decompose the 3D estimation problem into two steps, namely: (i) using a stacked hourglass network, return a simultaneous estimation of joint locations and a silhouette mask, and (ii), feeding this into two additional independent networks, estimate pose θ and shape β. After receiving outputs from the stacked hourglass, the pose parameter network passes 2D joint locations and their confidence values through fully-connected, bilinear layers, while the shape parameter network takes a human silhouette mask, filtering it through five convolutional layers before reducing dimensionality using fully-connected layers. We are able to be train these networks by capitalizing on the intermediate 2D inputs: we now have the ability to generate instances of the SMPL model with different 3D pose and shape, using the pose and body shape captures of real humans from existing datasets. From this, we can directly mask silhouettes and regress joints to get accurate ground truths.

3.2.5 Discriminators

One way to ensure robust results while reducing computation is to introduce a discriminator module [8], acting as a final gatekeeper of sorts. Like how the intersections of body parts are penalized in (8), a discriminator can be trained to tell whether any SMPL model parameters θ and β correspond to a real human or not. Rather than scouring the solution space in the optimization process, we can simply throw away a result if it is deemed unsatisfactory under the weak supervision provided by the discriminator.

3.3 End-to-End Methods

While multi-stage approaches carry more useful information in the intermediate stages, they often run sub-optimally, carrying out unnecessary or repeated computations, depending on the exact task and architecture of each subcomponent. End-to-end methods provide an alternative, since synthetic data can be created by generating instances of the SMPL model, and leveraging existing motion capture and human pose datasets. Kanazawa *et. al* [8] argue that end-to-end methods preserve global context, while avoiding the need for dual-stage training. They encode the image using ResNet-50, subsequently average pooling and regressing using two fully-connected layers, finally reducing dimensionality using a last fully-connected layer. Their method remains robust even when trained on unpaired 2D and 3D datasets.

3.3.1 Data preparation

Traditionally, pose and shape datasets come either in the form of images and annotated 2D keypoints, or 3D meshes and scans from motion capture sets. With only a standard marker set, MoSh estimates marker locations on a proxy 3D body model (in this case, SMPL), estimates the body shape, and recovers the articulated pose. By allowing body shape to vary over time, MoSh is able to capture the non-rigid motion of soft tissue. Motion capture datasets like CMU [1] and Human3.6m [7] can be MoShed to obtain ground truth 3D model parameters, like θ and β in SMPL. The network can then be trained on this mapping.

4 Conclusion

The end-to-end framework in [8] is able to run in real-time alongside a fast bounding box detector, YOLOv3 [14], running on two Quadro P5000 GPUs. In order to include hair, headgear and clothing deformations, we might need to look at a silhouette-based method like [2]. This gives rise to a trade-off between inference time and the ability to regress auxiliary shape. Texture mapping is also possible by back-projection of the model on several frames, then reading off the color values [2], or direct shape regression using a learned pixel-to-UV mapping [5]. The idea of using multiple, successive frames in a video to refine our estimates and avoid frame-by-frame computation might be feasible. A system built to recover full parametric model pose θ, shape β, offsets D, and texture maps, all in real-time, looks extremely feasible and will likely be the next step in this field of computer vision research.

References

[1] Motion capture database. CMU Graphics Lab. Funding from NSF EIA-0196217.

[2] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3D people models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] 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. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science. Springer International Publishing, Oct. 2016.

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.

[5] R. A. Guler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. *arXiv*, 2018.

[6] R. A. Guler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. *arXiv:1612.01202*, 2016.

[7] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, July 2014.

[8] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Regognition (CVPR)*, 2018.

[9] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[10] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[11] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016.

[12] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. Verona, Italy, 2018.

[13] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[15] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.