Automatic Estimation of 3D Human Pose and Shape from a Single Image

Hugo Dupré
MVA Master's degree
Ecole Normale Supérieure
Cachan, France
hugo.dupre@ensae.fr

Pierre Perrault
MVA Master's degree
Ecole Normale Supérieure
Cachan, France
pierre.perrault@outlook.com

Abstract

In this report, we present our "Object recognition and computer vision" class project on the subject of human pose and shape estimation, following Bogo et al. *[6] work. We first analyze their method on single images. Then, through experiments on instruction videos, we have tried to deduce some improvements using temporal consistency.*

1. Introduction

Estimating 3D pose and shape from a single human image is an interesting problem for many applications, and particularly with the spread of instruction videos on the internet, which in theory allows a machine to be able to learn how to perform difficult tasks. By $pose$ we mean the spatial layout of the human skeleton when the picture was taken, namely the articulate posture of the limbs in 3D. By $shape$ we mean the surface of body in 3D. It may seem very interesting to use the powerful feature space of a convolutional neural network (CNN) to tackle this problem. But they do not succeed in estimating pose and shape from one image. However, CNNs have been successful at estimating 2D joints locations of the human image: for example the DeepCut CNN ([9, 7]) reaches the state of the art in this task. Only using these 2D joints, Bogo *et al*. [6] proposed a new fully automatic method, called SMPLify, to compute both the body pose and shape. Our purpose in this project is to expose and experiment this new algorithm.

2. SMPLify method

We give here a short description of SMPLify. To summarize, they used a generative model of human bodies (i.e. pose and shape), called SMPL ([8]), and they optimize its parameters in order to match the model joints projection with the input 2D joints.

Figure 1. Example of female shape (left, taken from [8, 1]) and pose (right, taken from [1]).

SMPL SMPLify outputs an SMPL body (a triangulated surface with 6890 vertices) parameterized by both pose and shape. Shape is characterized by coefficients β of a PCA shape space (trained from thousands of registered scans). There are 3 models: male, female, and gender-neutral. Pose is represented by θ, the relative rotation between 23 body parts. From β, one can compute $J(\beta)$, the 3D skeleton joint locations, that can be put in any pose θ using a global rigid transformation R_θ: $R_\theta(J(\beta)_i)$ is the posed 3D joint i according to θ. Figure 1 illustrates an example of shape ($J(\beta)$ is represented by the white dots) and pose.

Objective Function To fit vectors (β, θ) to the 2D joints J_{est}, they minimize an objective function:

$$E(\beta, \theta, \Pi; J_{est}) = E_1(\beta, \theta, \Pi; J_{est}) + E_2(\beta, \theta) \quad (1)$$

Where Π is the projection from 3D to 2D induced by a camera. The first term, which is computed from the data, is a distance between J_{est} and corresponding projected SMPL joints:

$$E_1(\beta, \theta, \Pi; J) = \sum_{\text{joint } i} w_i \rho(\Pi(R_\theta(J(\beta)_i)) - J_{est,i}) \quad (2)$$

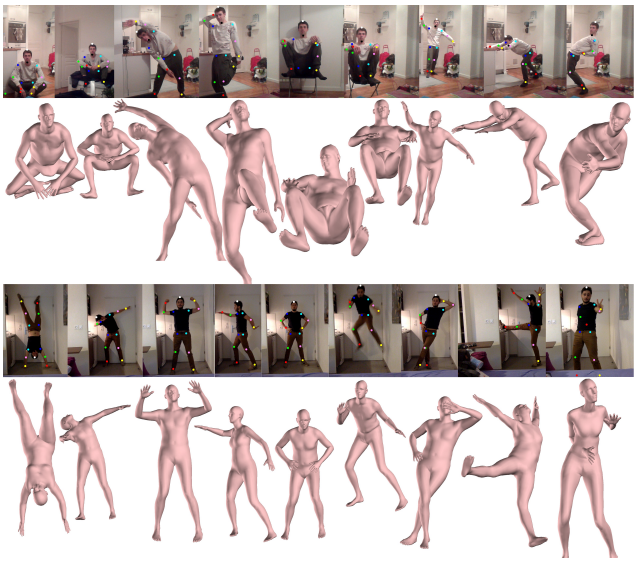

Figure 2. Results we obtained using pictures of ourselves. 18 poses (Up: Pierre and $Below$: Hugo) with corresponding DeepCut joints, and below SMPL bodies computed with SMPLify method.

Where w_i is the CNN confidence of $J_{est,i}$ and ρ is a differentiable penalty function. The problem of only considering E_1 is that there is no uniqueness of the minimum (β and θ can explain the same 2D image). Therefore, they consider a prior term, E_2, of which we only give an overview here; it is composed of 4 terms:

- A term corresponding to joint angle limits.
- A pose prior term trained from CMU dataset ([2]).
- An interpenetration error term: using an approximation of shape by capsules and penalizing the intersections between the latter.
- A shape prior term from SMPL PCA space.

3. Experiments

With our own images Using the code provide by [3], as well as the DeepCut CNN to compute the joints ([4]), we produced some results with our own images (see figure 2). Our poses are of varying difficulty. We first computed joints on each image (using different scales: 0.3,0.4 and 0.5; the algorithm then store automatically the most confident). Finally, we computed the bodies using the gender neutral model. On the one hand, we can notice that errors on joint computation are more rare when more scales are used, but that there are still some (for example the right knee of the first photo of Pierre is misidentify). Although the data term in the optimization is weighted by joint confidences, these joint errors are not well corrected on the next stage. On the other hand, results after the next stage are globally quite satisfying: 11 (6 for Pierre and 5 for Hugo) poses were optimally (knowing only joints) found. Errors are different:

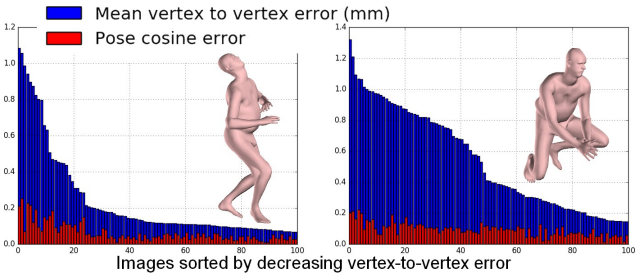

Figure 3. Evaluation on synthetic data. Left: standing bodies. Right: crouching bodies. Gender used is neutral.

the two last pose for Pierre and Hugo are interpenetrations, underlining the fact that the corresponding term in the optimization doesn't totally exclude interpenetration. Causes are not joints, but rather body part occlusions. Another kind of mistake come from unusual postures (see 4th picture of Pierre, the left knee was moved): even if joints are correctly identify, prior terms moved the knee seeing that the hip and ankle are close. Other mistakes come mainly from ambiguous joints where we do not know what body parts overlaps (see 3rd image of Pierre and 7th of Hugo) or which orientation has the whole body (see 2th and 8th image of Hugo).

Quantitative evaluation with synthetic data In order to highlight the sensitivity of the method to posture difficulty, we sample 100 SMPL standing bodies and 100 SMPL crouching bodies (using SMPLify on some frames from [5]). We then fix a known camera: from this point, we computed the ground truth 2D joints by projecting 3D joints into the image as in [6]. We then used SMPLify to compute SMPL bodies from these 2D joints. Figure 3 shows, for both kind of body, the pose cosine error[1] (red) and the mean vertex-to-vertex euclidean error between the estimated and true shape in this pose (blue). Notice that [6] use a canonical pose instead. We also compared averages of β euclidean error (see table 1 for averages errors).

Table 1. Mean errors for β, θ and vertex.

	β	θ	Vertex
Standing	0.6377	0.05873	0.2180
Crouching	0.9655	0.06836	0.3679

The results of shape estimation are more accurate for standing bodies, as we could expect. Moreover, one can see that our results are very similar to [6] ones (male and female all joints): it is not surprising seeing that pose error is quite low in both cases, and therefore shape are more or less compared in a same canonical pose.

[1] $\sum_{i=1}^{100}(1 - cos(\theta_{t,i} - \theta_{e,i}))$ where θ_t is the ground truth pose coefficient and $\theta_{e,i}$ is the estimated one.

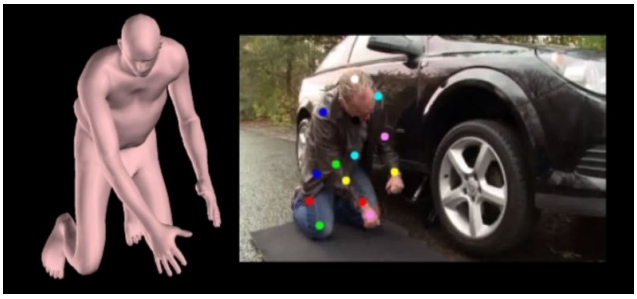

Figure 4. An example of frame from a changing tire video where the method works, with DeepCut joints and Resulted body we obtained using SMPLify.

With images from instruction videos We used instruction videos provided by Jean-Baptiste Alayrac ([5]). We consider videos as sequences of static frames and use the method on each frame. We experimented on small clip from `changing_tire_0012` (see figure 4), `repot_0018` and `cpr_0005`. Results obtained are very jerky and many bodies are not properly restored (see table 2), but we can still recognize general gestures of the actor. The most common errors come from one or two misidentified joints, and appear especially when some body parts are obscured.

Table 2. Imperfect body frame proportion on 3 clips from V1: changing_tire_0012, V2: repot_0018 and V3: cpr_0005.

Video	V1	V2	V3
Error proportion	0.3491	0.9642	0.3125

4. Improvement through temporal continuity

Last results we obtained on instruction videos can be improved using the fact that between two consecutive frames, results must not differ much. As soon as 2D joints are computed with DeepCut, one can notice that some joints in some frames are misplaced. Thus, we need to pre-process these joints. We detect outlier joints by computing their distance from those of the previous frame, i.e. the absolute velocity of joints, and then replace unconfident joints [2] having an absolute velocity greater than a certain threshold [3] by previous joint translated by α [4] time its velocity. By only doing this, we are facing a problem: if a disturbed joint was accelerating, the motion would not be well reconstituted. A good way to circumvent it is to then convolve the joints with a Gaussian filter [5]. We also have to ensure that the first frame joints are correct (to not propagate errors). We note that a high velocity does not necessarily correspond to an aberration. Perhaps considering acceleration instead is more relevant. However, since actors in our videos do not make abrupt moves, limit ourselves to velocity is resonable. Moreover, aberrant joints could have been interpolated more accurately by estimating higher derivatives of velocity. We experiment this joint processing on changing_tire_0012 (see Annex for other experiments on [5] videos). Figure 5 shows how this method removes misplaced right hip joints. If we use SMPLify on new joints, imperfect body frame proportion decreases to 0.05325 (9 frames over 169). Remaining errors no longer come from 2D joints but from SMPLify only. In order to eliminate these last errors, we had several ideas, which would be interesting to exploit in future works, for instance initializing SMPLify optimization with previous frame body, adding a continuity term in the optimization or detect pose aberation as we did for 2D joints.

[2] i.e. joint having DeepCut confidence less than a threshold (we choose on our examples 0.9)

[3] We took 50 for our examples, notice that it depends on the image size.

[4] We took $\alpha = 3/4$ for our examples.

[5] We took $\sigma = \sqrt{2}$.

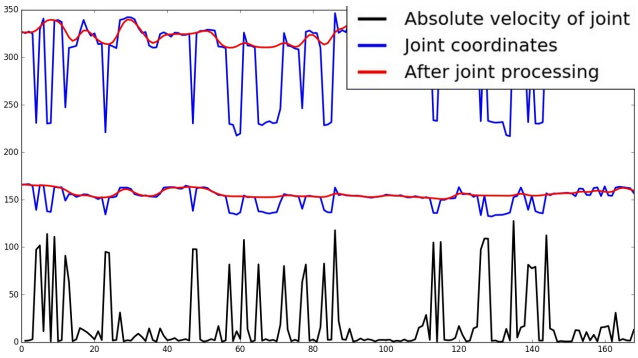

Figure 5. Right hip joint description across frames of changing_tire_0012 video before and after removal of aberrant joints and smoothing, as well as joint velocity used to detect aberrant joints.

5. Conclusion

We demonstrated the power and robustness of the SMPLify method in estimating 3D human pose and shape introduced by [6] through experimentations on images and an extension to videos, which provides very good results while remaining simple and using only 2D DeepCut joints.

References

[1] `http://smpl.is.tue.mpg.de`.

[2] `http://mocap.cs.cmu.edu`.

[3] `http://smplify.is.tue.mpg.de`.

[4] `https://github.com/eldar/deepcut-cnn`.

[5] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] 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.

[7] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schieke. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, 2016.

[8] 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.

[9] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

6. Annex

Here there are some frames showing improvements of joint processing. Full videos are available on Youtube (`https://youtu.be/DCZOwggVKD4`).