Paris Sciences et Lettres
École Normale Supérieure

Hugo Abreu & Anita Durr
CPES3 Informatique

Artificial Vision

Report - Human shape and pose

1 Introduction

Human shape and pose recognition algorithms can be useful for many applications, such as gaming, augmented reality, human-computer interaction, etc...

We will briefly present this subject through 5 five main articles, [1] [2] [3] [6] [7].

2 Problem

Two key characteristics of a human position are its pose and its shape. The shape is the general form of a person, it varies with its weight, its body proportions, its clothing, while the pose corresponds to the articulation of the body.

Input

We can determine shape and pose from two types of images: 2D images and depth images. The use of depth images varies from implementation to implementation. We will be looking at the implementation of Shotton et al., who uses images gathered from a Kinnect camera (the processing of depth images will be covered later).

Output

One way of determining pose (and some parts of the shape – height and proportions for example) is to modelize a human body in terms of its *joints* (also referred to as *keypoints*). They correspond to the points where two cohesive bodies (limbs) connect, and impose constraints on their relative movement. The set of joints of a human body determines the set of feasible movements it can perform. A set of joints is defined by the angle and distance between each join in a human body model.

Dense pose mapping, or the projection in the image of a human body segmented into different body parts (usually, corresponding to cohesive bodies) can be defined in different ways. It can be in 2 dimensions, containing information only on the body parts, or in 3 dimensions, with information on pose and the layout and relative position of limbs.

To modelize a whole human body (not just from the perspective of the image we are working on), we can use a *mesh representation* – a 3D model of the human body. A standard implementation of which is the Skinned Multi-Person Linear model (SMPL), introduced by Loper et al.. It is parametrized by 3D joint angles and shapes (represented in a low dimensional linear space). It is able to create an anatomically accurate model of the human body.

3 Datasets / Tools

Here are some usefull datasets and tools for human pose and shape recognition. We can use datasets with different types of labels, for different applications.

- **2D image → 2D joint**: commonly used datasets with annotated 2D keypoints are LSP, LSP-extended, MPII or MS COCO.
- **2D image → 3D joint**: To train algorithms to recognize 3D keypoints from a 2D image, the MoCap dataset can be used: it links 3D joint positions to their corresponding 2D images. MoCap actually provides videos, from which we can extract individual images.
- **2D image → 3D joints + dense pose mapping**:

  Güler et al. manually created a dataset, based on COCO, which adds labels pertaining to dense pose mapping of the human body: the DensePose-COCO dataset. This approach is very time consumming, and only an entity such as facebook (which has the man power to label such a large dataset: 50K images) could do it. In the article, the authors go into detail on how to build this dataset. It provides efficiency gains, as seen later.
- **2D image → 3D joints + SMPL**: MoSh is a method that given raw 3D MoCap marker data produces an SMPL mesh representation. It can be used upon a MoCap dateset (e.g. Human3.6M) to retrieve 3D joints or SMPL.

MoCap datasets, since they were created in laboratory conditions, don't necessarily reflect on real life images (referred to in the field as *in-the-wild*). Models trained with MoCap datasets can experience difficulties identifying body poses in images containing challenging conditions, such as occlusion, difficult viewpoint, crowded scenes, etc... It is important to choose a dataset according to what we are trying to predict.

One of the papers also proposes to create its own dataset, with synthetized images of valid randomized body positions. A dataset of 2D images associated with 2D or 3D joints can be created using MoCap and texture and background datasets (to compose a new image with random body positions: nevertheless, those body positions must be valid body positions, condition verified by comparing to MoCap examples). This approach is described by Varol et al. in [7].

4 Approaches

Since Neural Network (NN) algorithms require cohesive data structures to opperate efficiently, images used in the various experiments are cropped before being fed into the network (only selecting the zone of interest). In our case, we usually want to center the image on the body. Image size is also reduced for efficiency purpose.

Several NN architectures can be used in human pose recognition problems. One of the most common is the stacked hourglass architecture [5].

4.1 From a 2D image

The Iterative Error Feedback (IEF) is a powerful framework proposed by Carreira et al. in 2016. It does not directly predict the output pose, but determines the correction it has to make on its current prediction. The network is iteratively trained for a certain number of corrections of each image. By training a NN using IEF to predict 2D joints from a single annotated RGB image, the authors showed the advantage gains of using this method. This method was reused in several other papers pertaining to this topic, such as [3].

In-the-wild images can be difficult to analyse when they present difficult backgrounds, occlusions or scale variations. To handle these problems, Güler et al. trained a neural network with their DensePose COCO dataset, which - used in conjunction with an inpainting network - will fill in missing ground truth values in the first dataset and process the difficult conditions cited above. This allows the construction of a highly performing network that can run in real time on a single gpu and that can accurately predict human poses on *in-the-wild* images.

The classical approach to construct a full 3D mesh from a single 2D image is to first estimate the 2D joint locations and then to estimate the 3D model parameters (3D joint locations and shape). Kanazawa et al. propose instead to predict the parameters directly from the original image.

They use, for this purpose, a set of images with 2D joint annotations and eventually 3D joint annotations. The trained neural network has to infer the SMPL parameters, in a way such that the 3D joint projections resembles the 2D annotation. To make sure that the predicted parameters correspond to a plausible human configuration, they are also sent to another network – referred to as a *discriminator network* – which was trained to recognize real body SMPL using a large set of SMPL human bodies, in various poses and shapes.

4.2 From a depth image

Shotton et al. proposes a method of retrieving 3D positions of body joints from a single depth image, using no temporal information.

Using a single frame from a Kinnect Camera, they first define several localized body part labels that densely cover the body (adapted from MoCap classifications - in their paper: 31 body parts). The pairs of depth and body part images are used as fully labeled data for learning the classifier.

A randomized decision forests algorithm is then used to parse an image pixel by pixel (analyzing depth features, by comparing each pixel to offset ones) and find an ideal body part association for that pixel. This method is sufficient to accurately disambiguate between trained body parts.

To generate reliable proposals for the positions of 3D skeletal joints, the authors propose an implementation of a mean shift algorithm with a weighted Gaussian kernel to find the global 3D centers of probability mass for each body part: each of those centers of probability mass will form one joint, in the 3D skeletal model. These centers can be used, along with the original depth mapped image, to form a dense model that takes shape into account.

Since the analysis of depth features through the decision forests algorithm can be easily implemented in a massively parallel system (a GPU for example, even a mobile one), Shotton et al.'s approach can be easily used for real time human pose recognition (200 frames per second) on mobile hardware.

References

[1] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.512

[2] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7297–7306.

[3] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00744

[4] 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)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[5] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *Lecture Notes in Computer Science*, p. 483–499, 2016. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46484-$8_2$9

[6] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[7] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.492