Marker-less Pose Estimation

Andy Gilbert, Simon Kalouche, Patrick Slade
Stanford University
{adgil, kalouche, patslade} @stanford.edu *

Abstract

The ability to capture human motion precisely has benefits in various applications ranging from biomechanics studies, to physical therapy and exoskeleton control. The current most utilized methods for determining body kinematics and motion rely on either 1) optical systems which require retrofitting a room with an array of expensive cameras and tagging the subject with retro-flective markers at body locations of interest for tracking or 2) wearable inertial measurement units (IMUs) which require precise calibration and are tedious to don and doff. We propose to explore an alternative method, originally developed by Rhodin et. al in [10], which overcomes the limitations of current motion tracking solutions using a marker-less, body-mounted device consisting of two low-cost fisheye cameras. In this study we follow the methods of EgoCap, the work of Rhodin et. al using their publicly available dataset [10]. We train a ResNet model to get 2D joint heat-maps for 18 joint locations. The uses of a K-nearest neighbor and multi-layer perceptron are then compared for obtaining 3D pose estimation from the 2D joint heat-maps.

1. Introduction

Human motion capture is used in a variety of industries from film making to virtual reality, to prosthetic fitting and biomechanics studies. A particularly interesting application makes use of motion capture data to train deep neural networks on human motion during various activities to ultimately predict a human's intent of motion in real-time. Such a development can significantly improve upon the challenging control problem of synchronizing upper and lower-body exoskeleton devices with the human body to augment and assist natural motion rather than impede it.

However, collecting such a large dataset of human motion for a diverse set of tasks is currently difficult due to the limiting nature of currently utilized motion capture systems which include optical marker-based tracking, optical marker-less tracking and wearable inertial-based tracking. Optical systems offer very accurate tracking of many different points but these systems require outfitting a room with expensive IR cameras which must have overlapping fields of view in addition to the cumbersome and tedious task of placing uncomfortable retro-reflective markers on the human subject at several locations on their body. Optical systems also suffer from a limited capture volume constrained by the size of the room and do not work well in outdoor environments. Several optical motion capture algorithms have been developed using the marker-less approach and machine learning to estimate body-pose however these systems are similarly limited in their requirement of an external camera system and thus limited capture volume.

Inertial-based tracking offers advantages over optical marker-based motion capture systems in that it can be a wearable and thus portable system operating indoors as well as outdoors across a large spectrum of activities. Inertial-based marker-less motion capture however suffers from calibration and drift issues from the IMU's and requires N body-worn sensors to track the orientation of N joints or body segments.

Alternatively, a new approach called EgoCap proposes the use of two head-mounted fisheye cameras in a marker-less, optical inside-in motion capture system [10]. Using a pair of simply worn fisheye cameras and a trained deep learning model full-body human pose estimation can be achieved in real-time in indoor and outdoor environments over a diverse set of activities. In addition to achieving better results than IMU-based portable motion capture systems, EgoCap is able to also estimate global positioning using structure-from-motion on the scene background [10]. This method achieves capabilities of whole-body tracking analogous to Leap Motion's tracking of hand pose.

EgoCap's algorithm for whole-body motion capture is broken into several steps:

1. local skeleton pose estimation with respect to the body-mounted cameras

2. global pose estimation of the body-mounted cameras with respect to the world inertial frame

*This work was conducted in partial fulfillment of Stanford's CS 231n course: Convolutional Neural Networks for Visual Recognition.

In this study we propose building off of the original work of EgoCap by modifying the ResNet architecture and its hyper-parameters. The input to this network will be an image of a person from the EgoCap dataset and the output will be 18 predictions of different 2D joint locations. We also plan to explore several networks including separate K-nearest neighbor and multi-layer perceptron approaches to get the 3D body pose estimations. The input to these architectures will be the 18 joint location estimates in 2D and the output will be 18 joint pose estimates in 3D. This performance will be compared against the EgoCap ResNet model and their local skeleton pose estimation, achieved using an analysis-by-synthesis optimization which maximizes the alignment of a projected 3D human body skeleton model with the pair of images captured from the body worn fisheye cameras.

2. Related Work

The demand for accurate and efficient human pose estimates is increasing as platforms for human computer interaction (HCI), ambient computing, and biomechanical systems multiply (for both scientific and consumer applications). Pose estimation gives these computer systems the ability to interpret human intent and update their state based on this information. There are several different methods for pose estimation with techniques varying by application.

Generative models use a three dimensional computer model and attempt to match a two dimensional image to the known three dimensional model [16]. Generative models consist of two steps. In the first step a probability map is constructed based on known information such as the body model, camera type, and any image features. Second, the pose is estimated based off the probability map and any constraints. This is a frequent approach used with monocular pose estimation where the user only has an image from a single camera with which to obtain information. Without the depth information provided by biocular vision, having a prior model is crucial to accurate decoding. For human pose estimation the constraints would be those imposed by the limited motion of human joints. Burenius et. al present an example of a generative model. The authors develop a pictorial structure model (PSM), a representation of human body parts, for three dimensional reconstruction using a tree graph to connect the body parts and a Bayesian network to represent the relationships between connections [2].

Meanwhile, discriminative approaches start with no prior conception of the body and attempt to learn mappings between different joints. The pose is then estimated based on a series of training examples. Huang and Yang attempt this by evaluating the minimum linear combination of training samples that can be used to recreate a test sample [6]. The solution can be formatted as a convex optimization problem and solved quickly. Alternatively, Sedai et. al. cluster the three dimensional pose space into several regions and learn regressors for how to fuse different features in each region [15]. Generative models are generally more accurate and generalize better to complex poses, but are much more computationally complex due to their increased dimensionality [14]. Alternatively, generative and discriminative approaches can be combined into a hybrid approach where generative methods are used to enforce distance constraints in the discriminatory models [12].

Recently, deep learning has also gained popularity in pose estimation due to the power of convolutional networks in object recognition and classification.At first convolutional networks were only applied to two dimensional pose recognition [3] but recent work extended this to cover three dimensional pose reconstruction as well [8]. Convolutional network approaches have been successful but have suffered from a lack of sufficient training data due to the previously described difficulties of obtaining accurate three dimensional pose data [14]. To solve this problem several groups have tried to synthesize training images with annotations [4] [11]. Due to the difficulty of obtaining three dimensional data many groups take the approach of using two dimensional data sets to train the convolutional neural networks and then using a generative or discriminative method to implement a two dimensional to three dimensional transformation.

These methods can be implemented with either a monocular (single camera), biocular (dual camera), or multi-camera setup. Monocular setups have the greatest difficulty as cluttered backgrounds, occlusions, and ambiguity between two dimensional and three dimensional poses all present greater challenges with only one frame of reference. However, monocular setups also allow a more portable system as there is no need to continuously synchronize different images or maintain multiple cameras around a subject. Moreover, most of the data presently available is captured from monocular setups.

While we used a dataset collected with a biocular setup we were only provided data from one camera. Therefore, in our implementation we chose to use a convolutional net trained on monocular data that estimated two dimensional poses and then translate to three dimensions using a neural network as well as discriminative approaches.

3. Methods

Our approach at achieving 3D markerless motion capture of human body poses is accomplished via a 2 step process. First, the raw image from a camera or frame in a real-time video is fed into a very deep 101-layer residual network. The residual network (ResNet) learns a mapping between input 2D images ($H \times W \times C$) and pixel locations of each body joint of interest ($num\_joints \times 2$). The second step takes the output of the ResNet (i.e. the 2D (x,y) pixel coor-

dinates for each body joint) and feeds it into a multi-layer perceptron or neural network which learns a mapping from 2D pixel coordinates to 3D Cartesian coordinates in space. The 3D Cartesian coordinates correlate to the (x,y,z) locations of each body joint relative to a static global reference frame. We aim to improve the accuracy of this recognition process both in terms of the percentage of correctly identified joints and the distance from actual joint position based on the ground truth established by a network of outside-in cameras.

3.1. Image to 2D Pixel Coordinates

In order to train a body-part detector that will eventually allow for 3D pose estimation we first need to be able to predict body-part heat maps shown in Fig. 1. This will be accomplished by using a 101-layer residual network developed by [5] and used for pose estimation following the current approach [7].

ResNet uses network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping. These "residual blocks" contain two $(3x3)$ convolutional layers. Periodically the number of filters are doubled and downsampling is achieved with a stride of length 2. At the end of the residual blocks is a pooling layer and fully connected layers. These have been modified to output the 18 joint heat-map predictions.

For this project we used a modified version of ResNet as discussed in [7] from the code available at `https://github.com/eldar/pose-tensorflow`. They remove the pooling and final classification layers, decrease the stride of the conv5 bank of layers from 2 to 1 pixels to prevent downsampling, add holes to all (3×3) conv5 residual blocks to preserve the receptive field, and add deconvolutional layers to up-sample by two times and use the output of the conv3 bank as the actual output allowing for varying sizes of joint locations to be predicted. This makes the ResNet fully convolutional.

Figure 1. Body-part heat maps showing various joints in a sample image from the MPII dataset.

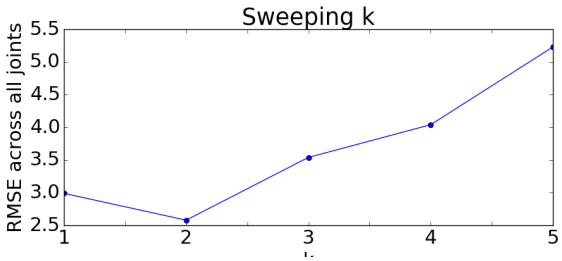

Figure 2. RMSE for different values of k.

3.2. 2D Pixel Coordinates to 3D Cartesian Coordinates

The overall 3D body pose accuracy will be determined by taking the average 3D Euclidean distance for all 18 points between the estimated values and the state-of-the-art distances found in [10] using multi-camera measurements. Two approaches were used to accomplish the second step of the total human pose estimation problem – going from 2D pixel coordinates to 3D Cartesian coordinates. The first approach is a K-nearest-neighbor (KNN) and the second is a multi-layer perceptron.

3.2.1 KNN

One way of translating from two dimensional joint data to a three dimensional pose reconstruction is a discriminative approach where coordinate transformations are memorized during training and the result is interpolated from the nearest examples during test, or a KNN approach. While this approach does not have any explicit representation of a body model, relying on an implicit from the collection of examples. In this case it has the advantage of training on images that all have the same frame of reference as those that will be presented during test. This was advantageous during this study as the setup prevented poses from varying a great deal between images. That is, the body was always positioned in the same relative space within the image. This was especially true of the head, neck, and hip joints, which varied only slightly across the image database.

Other studies have also shown KNNs to be optimal for 2D to 3D coordinate transformations with human pose data [11]. We subdivided the available 3D data into training and validation sets and experimented with the optimum value of k. We found that using the 2 nearest neighbors for reconstruction led to optimal performance. Results of sweeping k are shown in Fig. 2. The final RMSE for each joint is shown in Fig. 3. The algorithm does well for most joints, but struggles with smaller joints such as wrists and figures.

3.2.2 Multi-layer Perceptron

The multi-layer perceptron was built to experiment with network depth, number of hidden layer parameters, learn-

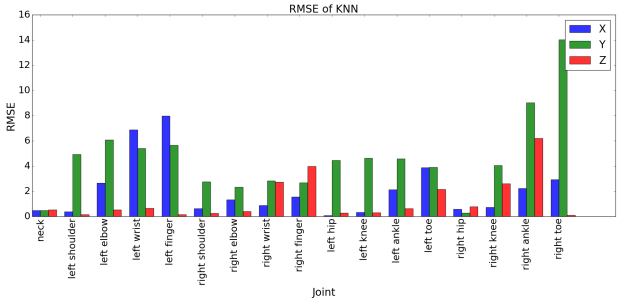

Figure 3. RMSE for each joint and each dimension.

ing rate, batch size, and dropout percent. The forward pass consists of repeating layers of affine fully connected layers, dropout, rectified linear unit (ReLU) non-linear activations, and batch normalization.

The network implementation is based off of open-source Tensorflow code: `https://github.com/aymericdamien/TensorFlow-Examples`. The input examples are structured in a randomly generated train and test set, X_{train} and X_{test} which are split from the total dataset using the 70-30 rule (70% of the data is used for training and 30% is used for testing). Since the EgoCap dataset had only 1000 labeled 3D examples our training set could use up to 700 examples for training and 300 for validation or testing. Therefore the shapes of X_{train} and X_{test} are $X_{train} \in \mathbb{R}^{700x36}$ and $X_{test} \in \mathbb{R}^{300x36}$ where 36 is the number of tracked joints (18 in this case) multiplied by 2 for the 2D pixel coordinates (x,y) corresponding to the row and column pixel location of each joint.

Tensorflow is used to implement the network layers as described above. The Adam optimizer is used to minimize the L2 loss defined by

$$L_2 = \frac{1}{n}\Sigma(h_\theta - y)^2 \tag{1}$$

where n is the number of training examples used in a single batch (i.e. the batch size), h_θ is the prediction, y is the label of size $1x3j$ and j is the number of tracked joints ($j = 18$ for EgoCap). The model is trained for 5000 epochs and hyper-parameters according to [9]. The best results attained used values of 0.001 for the learning rate, 60% dropout, top 2 layers each with 256 neurons and the bottom 2 layers with 1024 neurons, and a batch size of 64.

From Fig. 4 the root mean squared error can be seen to decrease steeply up through epoch 200 where it stabilizes to reasonable values. The RMSE then decreases slowly over the remaining 11,000 epochs but learning from this network architecture with the corresponding hyper-parameters saturates after approximately epoch 5000.

Adding additional layers to the network also did not seem to improve RMSE but it did significantly slow down the learning (i.e. increased training time).

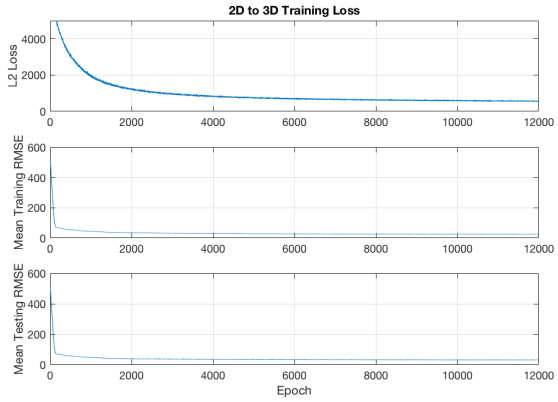

Figure 4. L2 loss, mean training RMSE, and mean testing RMSE versus training epoch.

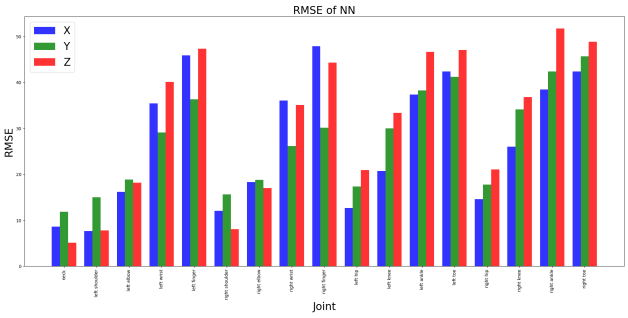

Figure 5. Mean RMSE after training 12,000 epochs per joint.

4. Dataset and Features

This work utilized the MPII Human Pose dataset [1] and the EgoCap dataset [10]. The MPII Human Pose dataset is comprised of 25k images with over 40k people performing various activities. Following the methods of [7] this dataset is preprocessed by locating the people and cropping images to focus on the individuals, generating a dataset of 42k images of people performing various activities. The 2D image pixel locations of 14 joints are given as labels for the training and validation images. They also provided framework code for initializing a ResNet model and training the fully connected layers using their provided data on their github: `https://github.com/eldar/pose-tensorflow`. Since the examples are used purely for learning additional features that correspond to joint heatmaps no validation set was utilized.

The EgoCap dataset [10] is a a series of images taken from video recorded on two cameras with fisheye lenses mounted to a helmet. The fisheye lens was chosen for its wide viewing area, attempting to minimize the amount of occlusion caused by leg or arm movements. The dataset contains 20k raw green-screen images, 75k augmented images with variations in background texture and user clothes color, and a 3D dataset recorded with a motion capture system. Augmentation was achieved by recording on a green

Figure 6. Example image from the augmented EgoCap training set.

screen and then substituting random images as shown in Fig. 6. This was implemented to prevent over-fitting. The images were labeled with 18 joint locations in the image space. A script was written to take the joint locations given in the EgoCap labels to reformat that for the DeeperCut ResNet framework, so it could be used for training and validation. The ResNet network was modified to train on the ResNet images to output 18 joint locations.

5. Experiments

These networks will be analyzed to look at performance metrics and qualitative examples of classification to understand the performance and error cases. We will evaluate our network based on two metrics. The first is simply its accuracy in learned body part detection. Our metric will be percentage of correct keypoints (PCK). The second metric will be the distance from predicted and actual joint positions based off the generated heat maps.

5.1. Image to 2D Pixel

The ResNet is trained in two stages. The initial training was done following the procedure in [7]. Where the ResNet was initialized with ImageNet-pre-trained models and then learned joint heat-map features on the preprocessed MPII dataset. The network was then trained on the EgoCap augmented dataset. A hyperparameter search was performed, initially just for the learning rate over the first 5000 iterations. A value of 0.0023 was found to perform the best, similar to the value of 0.002 used in [10]. The weights were initialized with the pre-trained models. The EgoCap training procedure was used with a batch size of 1 for initial training iterations of 200,000 and then the learning rate was dropped to the suggested 0.0002 value for 20,000 additional iterations. Stochastic gradient descent was implemented and the training images were randomly scaled by $\pm 15\%$ to make the training more robust for various sized users.

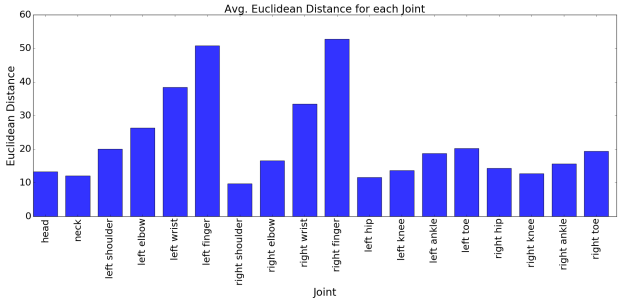

Figure 7. Average Euclidean distance between predicted and actual joint locations.

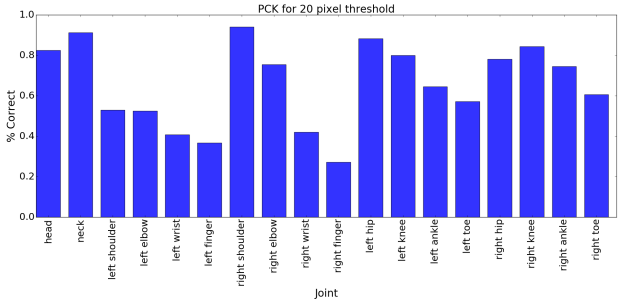

Figure 8. Percentage of Correct Keypoints (PCK) based on a 20 pixel threshold.

The accuracy of the body-part detection will be found with the percentage of correct keypoints (PCK) method [13, 17] following the validation parameters in [10] for a 20 pixel threshold shown in Fig. 8. The PCK is evaluated as the percentage of trials where the euclidean pixel distance between the actual and predicted joint location shown in Fig. 7 is below the desired threshold.

Results obtained in [10] showed a classification accuracy of between 60% and 90% on PCK for select joint locations. This is roughly 10% to 20% higher than our PCK values for the same threshold. Due to the parameter similarity in training, we believe this difference to be due to our difference in learning rate during training. It should be noted that the PCK for the joints on the leg is significantly higher than the arms, this is due to the relatively small size of the legs, making it easier for the network to be within the pixel threshold. A potentially better metric could scale the distance away the predicted value was by the size of the feature. PCK for all joint locations from the EgoCap tests aren't given, likely due to the high correct value regardless of threshold for other body parts that are mostly stationary relative to the camera frame.

Fig. 9 shows the 2D joint predictions on a raw EgoCap photo. The accuracy of the predictions is visibly worse on the limbs and locations further from the head as they move

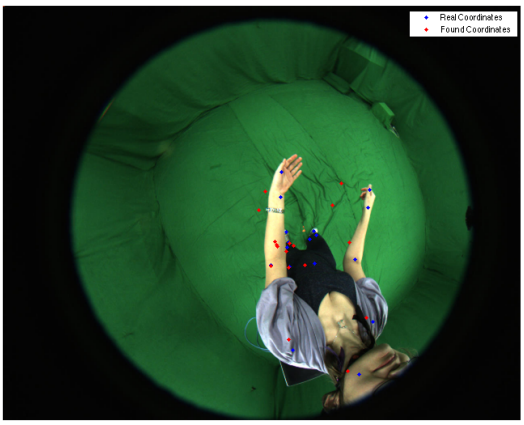

Figure 9. Accuracy of 2D predictions for an example image.

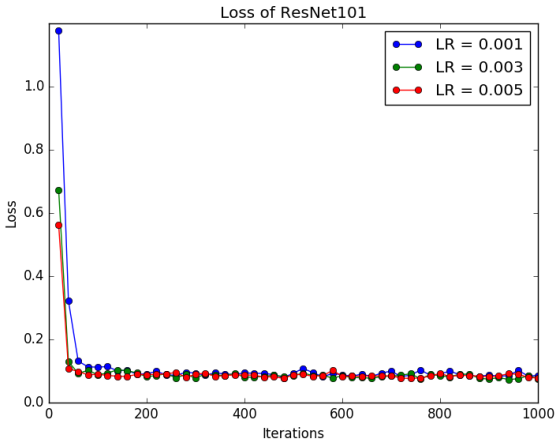

Figure 10. Learning rate hyperparameter tuning for a small number of iterations.

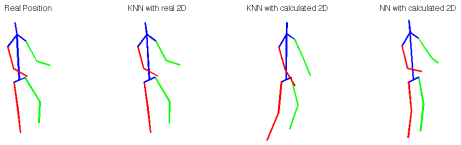

Figure 11. 3D pose reconstructions for example given in Fig. 9. 3D modeling code modified from `https://github.com/flyawaychase/3DHumanPose`

through greater ranges of motion, offer greater possibility for occlusion, are smaller due to distance, and are more skewed by the fisheye lens distortion.

5.2. 3D Pose Estimations

Two approaches were used to estimate the 3D body pose from a given or estimated set of 2D pixel locations on a single image. The KNN achieves a mean RMSE of XXX mm while the neural net receives a mean RMSE of 30.5 mm. While the KNN clearly seems to outperform the neural net (multi-layer perceptron) the KNN is certainly overfitting to the particular EgoCap dataset and will likely not generalize well to datasets and body poses not explicitly seen in the training set. Since the original EgoCap authors used a separate approach described as an analysis by synthesis optimization method which maximizes the alignment of a human body skeleton with 2D joint pixel locations the accuracies can be compared. While the optimization method proposed by the EgoCap authors is more accurate and robust to various unseen human body poses the robustness to changes in body type and dimension or length of limbs mull severely affect the accuracy of a model trained on a different sized person. As compared to the optimization method the KNN runs slower in real time due to the KNN iterating and calculating matches between the entire training set and the test example.

Additionally, Fig. 4 shows that the right side limbs have consistently higher root mean squared error in position estimation as compared to the left-side limb joints (arm and leg). This may be due to the fact that the input to the ResNet was the image taken from the left head mounted fisheye camera which has less occlusion to the body's left side limbs and joints as compared to the occlusion on the right side of the body. The EgoCap dataset only provided labeled data for the left side camera images. To improve the model, both the left and right cameras can be used as input into the model and a weighted average can be used to determine the final 3D Cartesian coordinates of each body joint.

6. Conclusion

This work highlights how a ResNet model can be trained to perform heat-map predictions for 18 joint locations from 2D images of a person's body. For extending this to a 3D pose estimate we found a KNN network is more accurate than a multi-layer perceptron network. Combining these prediction and pose estimation networks results in a method for performing 3D marker-less pose estimation with centimeter level of accuracy.

Future work could extend the validation of the KNN and multi-layer perceptron network to test on various subjects performing activities from data outside the EgoCap dataset to see if there is really any brittle over-fitting in the KNN model, or if the perceptron network can improve in performance relative to the KNN. This research can be extended by testing the 2D to 3D approach to live video feed taken while performing tasks. The method should be extended to adapt to different sized subjects rather than the select few present in the EgoCap data.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] M. Burenius, J. Sullivan, and S. Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3625, 2013.

[3] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3063–3072, 2016.

[4] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 479–488. IEEE, 2016.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[6] J.-B. Huang and M.-H. Yang. Estimating human pose from occluded images. *Computer Vision–ACCV 2009*, pages 48–60, 2010.

[7] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele.

[8] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2848–2856, 2015.

[9] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. volume abs/1705.03098, 2017.

[10] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 35(6):162, 2016.

[11] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in Neural Information Processing Systems*, pages 3108–3116, 2016.

[12] M. Salzmann and R. Urtasun. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 647–654. IEEE, 2010.

[13] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3681, 2013.

[14] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016.

[15] S. Sedai, M. Bennamoun, D. Q. Huynh, and P. Crawley. Localized fusion of shape and appearance features for 3d human pose estimation. In *BMVC*, pages 1–10, 2010.

[16] C. Sminchisescu. *Estimation algorithms for ambiguous visual models: Three dimensional human modeling and motion reconstruction in monocular video sequences*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2002.

[17] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.