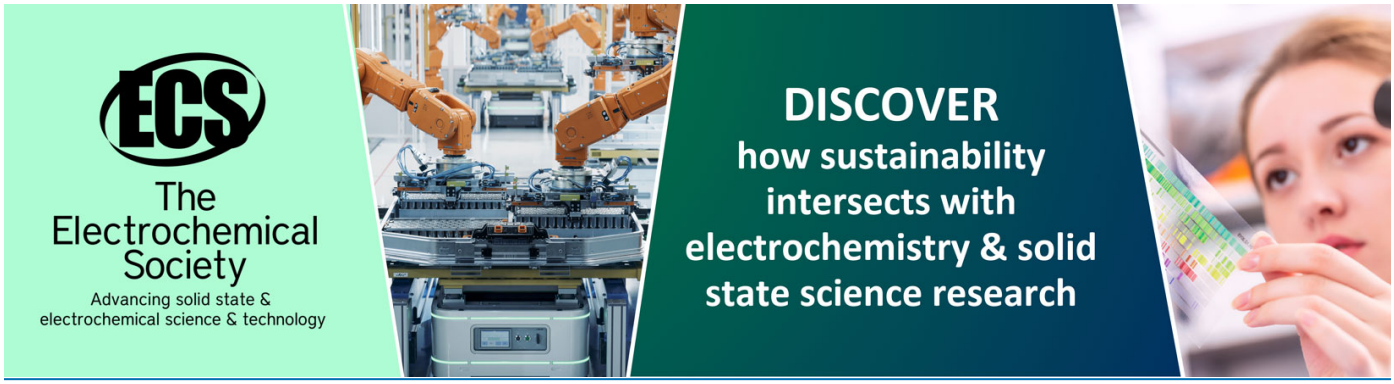
Visual System for Tracking Specific Human Body Extraction of 3D Skeletal Points Based on Monocular



IOP Conf. Series: Materials Science and Engineering **646** (2019) 012056 doi:10.1088/1757-899X/646/1/012056

Visual System for Tracking Specific Human Body Extraction of 3D Skeletal Points Based on Monocular

Baopeng Xu and Shuying Zhao*

Northeastern University, China

*568450666@qq.com

Abstract. In this paper, we introduce the Extracting specific human 3D skeleton point system based on monocular tracking. The system mainly consists of two parts. The first part is the detection and tracking of specific human body. This article uses simple online and real time tracking with a deep association metric (DEEP SORT)[1] algorithm, which is simple but effective, and meets system requirements in terms of efficiency and real-time. The second part is to extract the 3D bone points for the specific target of the tracking. We refer to Xingyi Zhou's research work[2] in this area. Utilizing the correlation between 2D pose and depth estimation subtasks, the training is end-to-end, and the algorithm introduces 3D geometric constraints to normalize 3D pose prediction, which is effective without ground truth value depth labels. In this paper, the two methods are combined by improvement, and the Extracting specific human 3D bone point system based on monocular tracking is designed. It can realize the tracking of 3D skeletal points of specific targets. The system has high practical value in human-computer interaction, virtual reality and motion recognition.

1. Introduction

The problem of human pose estimation has been extensively studied in computer vision. It has many important applications in human-computer interaction, virtual reality and motion recognition. The existing research work is divided into two categories: 2D pose estimation and 3D pose estimation. Due to the availability of large-scale 2D annotated human poses and the emergence of deep neural networks, 2D human pose estimation problems have recently achieved great success[1][3]. State-of-the-art technology enables accurate predictions in a variety of settings. The extraction of 2D bone points has made great progress, but the correct rate of understanding and analyzing human-behavior when extracting 2D bone point information is still not very high, which prompts us to have three-dimensional space for human bone points. The research of information and the practical application are more appropriate. In [4], it is also confirmed that the use of human body 3D skeletal point information is much higher than the correct rate of using 2D skeletal point information. In terms of human-computer interaction, the three-dimensional skeleton point information of the human body can be accurately obtained, so that the robot can directly respond to human instructions. In this paper, we extract the 3D skeletal points based on the information of the two skeletal points, and label the detected people to distinguish them to achieve the analysis of specific individual situations in the actual application process. It also meets the needs of human-computer interaction. In the previous computer vision to obtain three-dimensional information, they generally need binocular camera, multi-camera camera, or through the method of structure from motion (SFM)[5]. In our system, we use the

deep neural network to obtain the 3D bone points of the human body through the monocular camera. The requirements for visual equipment are simpler.

2. 3D Skeleton point extraction module

For our mission, our proposed network architecture as shown in figure 1.

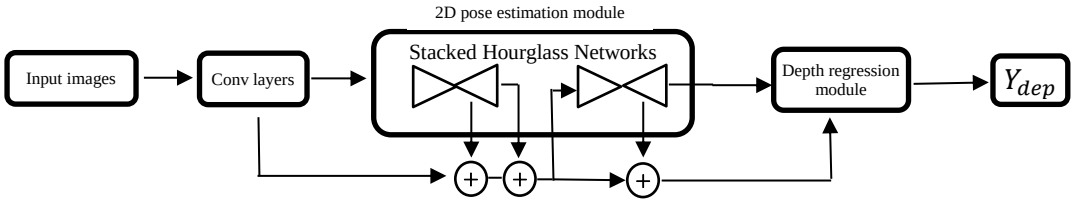


figure 1. Proposed network architecture

Here is a shallow stacking hourglass[1] model with stack=2. The deep regression module contains 4 consecutive residual and regression modules that can be considered as half hourglass.

The network is trained in a two-part datasets, the first part of images in the wild with only 2D ground truth, the second part of images in the lab with 3D ground truth.

2.1. 2D Skeleton point Estimation Module

Here we use the stacked hourglass [2] model as our system's 2D bone point extraction module, the network output is J low-resolution heat map[3], J is the number of joints. Each heat map represents the probability distribution of a joint. The peak position on these heat maps is the coordinate position of the 2D joint point we want to predict. Because these heat maps are also easily integrated with other deep layer feature maps, they serve as input to deep regression module.

To train this module, the loss function is:

$$L_{2D}\left(\hat{Y}_{HM}, \hat{Y}_{2D}\right) = \sum_{h}^{H}\sum_{w}^{W}(\hat{Y}_{HM}^{(h,w)} - G(Y_{2D})^{(h,w)})^2 \quad (1)$$

The loss measures the L^2 distance between the predicted heat-maps Y_{HM} and the heat-maps $\mathrm{G}(\mathrm{Y}_{2D})$ rendered from the ground truth Y_{2D} through a Gaussian kernel [3].

2.2. Depth Regression Module

If only the position of the 2D joint point is used as the input of the depth prediction, and then the training is performed according to the corresponding 3D annotation in the datasets, but if you do this, the result is not certain. because, in general, a single 2D skeleton has multiple 3D interpretations, so as above Here, we combine the J heat maps and other deep layer feature maps output by the 2D bone point extraction module by upsampling resize as input to the 3D depth regression module. J represents the number of joints. These features, which extract semantic information at multiple levels for 2D pose estimation, provide additional cues for 3D pose recovery.

According to Zhou et al[2] research, a new loss caused by geometric constraints[6] is proposed for weakly labeled data sets. In the absence of a ground truth depth label, this geometric constraint is used as an effective regularization of depth prediction. It is based on the fact that ratios between bone lengths remain relative fixed in a human skeleton (e.g., upper/lower arms have a fixed length ratio, left/right shoulder bones share the same length).

Specifically, let R_i be a set of involved bones in a skeleton group i , Here we consider four sets of bones: R_{arm}={left/right lower/upper arms}, R_{leg}={left/right lower/upper legs} , $R_{shoulder}$={left/right shoulder bones},R_{hip}={left/right hip bones}.

We let L_e represent the length of the bone e, and $\overline{L_e}$ the normal bone length. This value is set to the average of the dataset Human3.6M used for training. The ratio of each bone $\frac{L_e}{\overline{L_e}}$ in each group R_i should be maintained. So we set the loss function of the geometric constraint of the bone to the sum of

variance among $\{\frac{L_e}{L_e}\}$ e$\in R_i$:

$$L_{geo}\left(\hat{Y}_{dep}\middle|\mathrm{Y}_{2D}\right)=\sum_i\frac{1}{|R_i|}\sum_{e\in R_i}\left(\frac{L_e}{L_e}-\bar{r}_i\right)^2 \quad \text{Where } \bar{r}_i=\frac{1}{|R_i|}\sum_{e\in R_i}\frac{L_e}{L_e} \tag{2}$$

The length of the bone is a function of the position of the joint, which in turn is a function of the predicted depth. Therefore, L_{geo} is continuous and differentiable with respect to Y_{dep}. This allows us to continuously optimize the geometric loss. The loss of the depth regression module is:

$$L_{dep}\left(\hat{Y}_{dep}\middle|I\ ,Y_{2D}\right)=\begin{cases}\lambda_{reg}\left\|Y_{dep}-\hat{Y}_{dep}\right\|^2, if\ \ I\in\gamma_{3D}\\ \lambda_{geo}L_{geo}\left(\hat{Y}_{dep}\middle|Y_{2D}\right), if I\in\gamma_{2D}\end{cases} \tag{3}$$

2.3. Training

We combine equations (1), (2), and (3) as the loss function of training:

$$\mathrm{L}\left(\hat{Y}_{HM},\ \hat{Y}_{dep}\middle|\mathrm{I}\right)=L_{2D}\left(\hat{Y}_{HM},Y_{2D}\right)+L_{dep}\left(\hat{Y}_{dep}\middle|I\ ,Y_{2D}\right) \tag{4}$$

Since the network consists of two modules and there is a highly nonlinear nature that the geometric constraints cause loss, we divide the training process into three steps. In the first phase, we use the 2D dataset to train the 2D bone point extraction module. The second phase uses the 2D and 3D datasets to initialize the 3D bone point extraction module and fine tune the 2D bone point extraction module. At this step we do not enable geometric constraints. The third phase activates the geometric constraints.

2.4. Datasets

MPII-training. MPII dataset[7] is a large field dataset, image data is obtained from online video, and artificially annotated 16 2D joint points in the image. The data set contains 25,000 training pictures and nearly 3,000 verification pictures, and the human body has a border annotation in the image.

Human-3.6M. Human 3.6M datase[5] is widely used in 3D human pose estimation. This data set contains 3.6 million images captured by the motion capture system in an indoor environment. We reduce redundancy by interval sampling, according to the standard scheme uses five subjects (S1, S5, S6, S7, S8) as training set and (S9, S11) as our test[8] .

2.5. An exam Result display

An exam Result display as shown in figure 2.

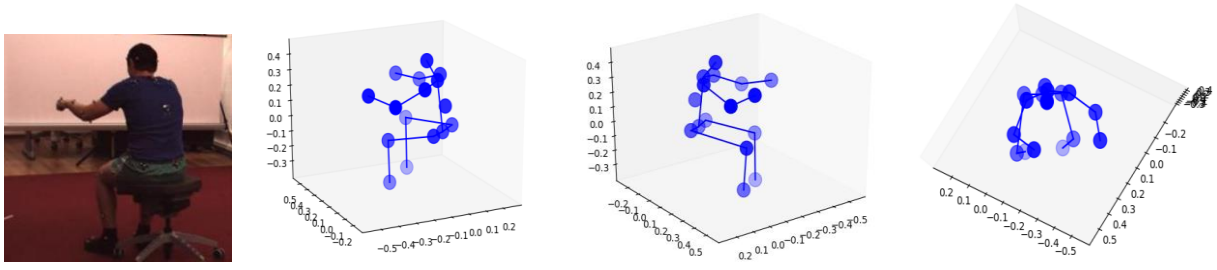


figure 2. *3D skeleton point extraction results*

3. Tracking Module

3.1. Track Handling and State Estimation

For our tracking module, we draw on the DEEP SORT algorithm[9], which uses a standard Kalman filter with a constant velocity motion and a linear observation model. The target we need to track contains the center position of the bounding box (u; v), the aspect ratio γ , height h and their

corresponding velocity in image coordinates. Among them, we use the boundary coordinates (u, v, γ, h) as the direct observation of the object state.

For a target a we want to track, we first set a threshold Amax and then calculate the number of frames since the last successful measurement of the association a. This counter is incremented during the Kalman filter prediction, when the target of the tracking is associated with the predicted amount. Reset to 0. But when the value of the counter is greater than the threshold Amax, means our mission target has left the scene we observed. Then remove from the tracking target collection. For every detection target that cannot be associated with an existing tracking target, we set these targets as tentative targets in the first three frames, and delete those targets if they are not successfully associated with the predicted values in these three frames. We need to explain that the traditional way to resolve the correlation between predicted Kalman state and newly arrived measurements is to use the Hungarian algorithm to solve the allocation problem. We use a combination of two appropriate indicators to integrate motion and appearance information and establish judgment conditions.

In terms of motion information, we can use the Mahalanobis distance between the predicted Kalman state and the new arrival amount to determine whether or not to correlate.

$$D^m(i,j) = (d_i - y_i)^T S_i^{-1}(d_i - y_i) \tag{5}$$

Mahalanobis distance away from the standard deviation of the average track position estimate how to consider the state detected by measurement uncertainties. We need to set a threshold for the Mahalanobis distance to determine whether it is related. This threshold is calculated by calculating the inverse $\gamma 2$ distribution when the confidence interval is 95%. For our four-dimensional measurement space, the corresponding Mahalanobis threshold is 9.4877. If the association between the i-th track and the j-th detection is acceptable, the evaluation is 1.

$$b_{i,j}^{(1)} = 1 \ \ if \ D^m(i,j) \le t^{(1)} \tag{6}$$

When the motion state of the target is relatively stable, the combination of Kalman prediction and Mahalanobis distance is a good correlation measure, but the prediction state derived from the Kalman filter frame is only a rough estimate of the target, when occlusion occurs or this metric becomes very inaccurate when the movement is unstable. So we introduce the appearance descriptor r_i and measure the minimum cosine distance between the i-th track and the j-th detection. This requires us to save a gallery $R_k = \{r_k^{(i)}\}_{k=1}^{L_k}$ of the last Lk = 100 associated appearance descriptors for each track k.

$$D^{cos}(i,j) = \min\left\{1 - r_j^T r_k^{(i)} \middle| r_k^{(i)} \in R_i\right\} \tag{7}$$

We also set a threshold for the cosine distance to determine if it is related.

$$b_{i,j}^{(2)} = 1 \ if \ D^{cos}(i,j) \le t^{(2)} \tag{8}$$

Appearance descriptors are derived from convolutional neural networks, In simple terms, using a convolutional neural network to extract the multidimensional feature vector of the target. so the appropriate threshold is obtained from the training set. We will introduce the structure of specific network in the following sections.

The Mahalanobis distance indicator determines the possible object position information based on motion, which is very effective for short-term prediction. The cosine distance is an indicator that takes into account the appearance information. If the motion state is unstable, the appearance information is useful for recovering the identity after occlusion. We use the weighted combination of these two metrics.

$$C_{i,j} = \lambda D^m(i,j) + (1-\lambda) D^{cos}(i,j) \tag{9}$$

The algorithm takes into account that when the target we are tracking is occluded for a period of time, the Kalman filter adds more uncertainty to the target prediction, especially when the two targets that are being tracked compete for the same detection, the Mahalanobis distance indicator will How to add more uncertainty, so a matching cascade was introduced to solve this problem.

As input we provide the set of track T and detection D indices as well as the maximum age Amax. Then calculate the association cost to select the set that can be associated, then we iterate the age

Amax to solve the linear assignment problem in the age increment. Next, the tracking target set Tn associated with the detection in the last n frames is selected, next we solve the linear assignment between the tracking target and the mismatch detection in the set Tn, and finally update the matched and unmatched detection sets and return.

3.2. Deep Appearance Descriptor

According to the above, the appearance descriptor of the algorithm is obtained by a neural network trained by[10] a large pedestrian recognition data set containing 1.1 million images of 1261 pedestrians. The advantage of using deep neural networks is that they can be accelerated with GPU, which is very suitable for online tracking.

The neural network applied in the algorithm first accesses two convolutional layers and one maximum pooling layer, followed by six residual blocks, and maps the extracted 128-dimensional global features to the dense layer. It fits the variables we need to calculate the cosine distance.

3.3. Result display

Result display as shown in figure 3.

Figure 3. Tracking module tracking results

4. 3D Skeleton Point Extraction Modul and Tracking Module Fusion

Our main goal is to take advantage of the capabilities of the two modules above to achieve our task of tracking specific targets and extracting their 3D bone points. For our follow-up work, it provides the basis for the action recognition, human-computer interaction and other work of specific targets.

The system uses YOLOV3[8]to detect the human body. But there are two issues that need to be faced in the process of convergence. One is that the target extracted by our tracking module returns in the form of a bounding box, but the border we get sometimes cannot completely contain the target. We have statistics for all test sets, and the tracking module the result of the operation, manual labeling measurement, and the curve fitting, the initial solution is to expand the length and width of the border to the original 1.126 according to the original aspect ratio of the extracted bounding box.

The second problem is that the general shape of the human body is rectangular, and the picture input by our 3D extraction module is fixed 256 × 256 size. This can cause serious deformation of the human body and cause huge errors. The solution we take here is to scale or expand a long side of the bounding box to 256 pixels according to the original bounding box aspect ratio, and to fill or expand the short side with less than 256 pixels to complement the white pixels.

We combine the two modules described in this article to build our visual system for tracking specific human body extraction of 3D skeletal points based on monocular. The overall work flow chart of the system as shown in figure 4

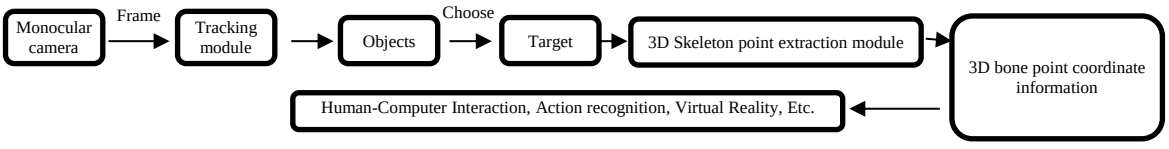
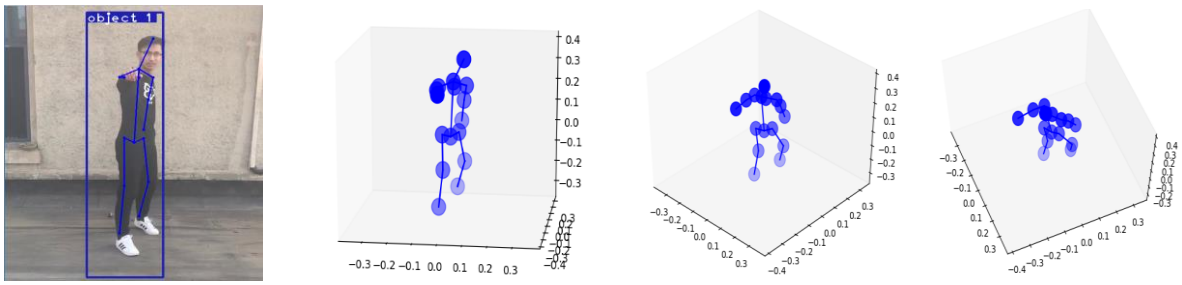


Figure 4. The overall work flow chart of the system

5. Experiment And Discussion

5.1. Single target tracking fusion experiment

Single target tracking fusion experiment as shown in figure 5

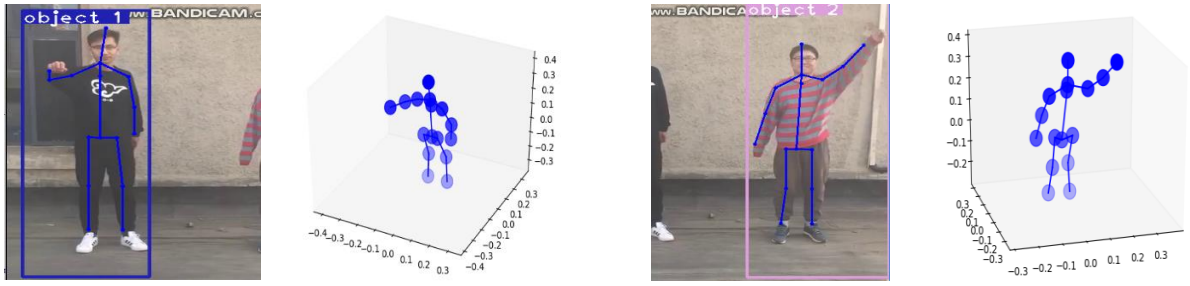


(a)Input (b) Front view (c) Side view (d) Top view

Figure 5. Single target tracking extraction skeleton point

5.2. Multi-target tracking fusion experiment

Multi-target tracking fusion experiment as shown in figure 6.



(a) Object1 (b) Object2

Figure 6. Track target with Object1 and Object2 and extract bone points

5.3. Discussion

At present, there are still many improvements in our system. First of all, the network of our system is not end-to-end. The next work can use the extraction module to extract the network of apparent information and the extraction features used by 3d skeleton points. The network is combined to strive for end-to-end training. Currently we are using single frame prediction, and we have discarded important time information. The next work can consider how to use the information in time. Another point is that our system needs to be configured to run on our robots. We can consider deploying a large amount of computing to the cloud processor while the system is running. This is also the trend of the development of robots.

Acknowledgment

Thanks to Professor Zhao Yuying for his careful guidance during the research process, and also to Xin Wu, Zhichhao Si for their help in the data collection and statistics section.

References

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. 2016. Simple online and realtime tracking. In 2016 IEEE Int. Conf. on Image Proces. 3464–3468

[2] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. 2017. Towards 3D human pose estimation in the wild: A weakly-supervised approach. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh. 2016. Convolutional Pose Machines. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 4724-4732.
[4] Sijie Yan, Yuanjun Xiong, Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. Thirty-Second AAAI Conference on Artificial Intelligence.
[5] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. 2009. From structure-from-motion point clouds to fast location recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[6] Akhter, I., Black, M.J. 2015. Pose-conditioned joint angle limits for 3D human pose reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 1446–1455.
[7] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2014. 2d human pose estimation: New benchmark and state of the art nalysis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[8] Li S., Chan A.B. 2015. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In: Cremers D., Reid I., Saito H., Yang MH. (eds) Computer Vision -- ACCV 2014. ACCV 2014. Lecture Notes in Computer Science, vol 9004. Springer, Cham.
[9] N. Wojke, A. Bewley, and D. Paulus. 2017. Simple Online and Realtime Tracking with a Deep Association Metric. CoRR, abs/1703.07402.
[10] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, Christoph Bregler. 2015. Efficient Object Localization Using Convolutional Networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 648-656.