# Fast 3D Post Estimation of Human Based on Optical Flow and Particle Filter

Jiaying Lan, Guoheng Huang*, Lianglun Cheng
School of Computer Science, Guangdong University of Technology, Guangzhou, Guangdong, China.

* Corresponding author. Tel: +86 13560231128; mail: kevinwongdh@gdut.edu.cn
Manuscript submitted May 31, 2019; accepted August 27, 2019.
doi: 10.17706/jsw.14.10.437-448

**Abstract:** Aiming at the high computational complexity of the 3D pose estimation algorithm in the deep learning field, we propose a fast human pose estimation algorithm based on optical flow and particle filter. The temporal correlation between video frames is applied to the algorithm. The first frame of the video is defined as a keyframe, which will serve as the output of the 3D pose estimate. Then the next frame is determined by the key frame algorithm whether it is a key frame. The key frame is estimated by the 3D human pose estimation algorithm, and the output result of the key frame is propagated to the non-key frame through the optical flow mechanism. Non-key frames are subjected to pose estimation through particle filter. In the 3D human pose estimation problem, we propose a unified equation for 3D human pose estimation from the RGB image, combining 2D joint estimation and 3D pose reconstruction. The proposed approach outperforms all state-of-the-art methods on Human3.6m achieving a relative error reduction greater than 30% on average. Our method significantly improves detection performance compared to the original algorithm, and the detection speed can be increased by an average of 43.75%.

**Key words:** 3D pose estimation, optical flow, multi-subspace particle filter.

## 1. Introduction

Estimating 3D human pose from an image is one of the most challenging problems in computer vision. It involves dealing with two challenging tasks. First, the 2D position of a human joint or landmark must be found in the image, resulting in a dramatic change in visual appearance due to different camera angles, external and self-occlusion, or changes in clothing, size or lighting. Next, lifting the coordinates of the 2D landmark from one image to 3D is still an ill-posed problem—the space in which the possible 3D human pose coincides with the location of the human 2D landmark is infinite. To find the correct 3D human pose that matches the image, it is often necessary to inject additional information in the form of 3D geometric pose a priori and time or structural constraints.

At present, the human body posture field is booming, which promotes the further development of human body posture detection. Among them, Cao *et al*. [1] performed a regression analysis on the joint points of the human body posture, and simultaneously used a parallel multi-stage network to extract the affinity field between the joint points of the human body, thereby determining the relationship between the joint points of the multi-person posture. Pfister *et al*. [2] used the optical flow information to distort the human body heat map of the front and rear frames to the current frame, and then assigned different weights to the heat maps at different times, and comprehensively obtained the current frame human body posture. He *et al.* [3] first used the neural network to extract the candidate region of the human body, and then used two parallel

networks to perform target detection and human joint detection on the candidate region. Pavloakos *et al*. [4] proceeded from a single color image through a convolution network (ConvNet) for 2D joint positioning and subsequent optimization steps to restore 3D human pose.

The above mentioned method can only solve the problem of human pose estimation for a single image. However, these methods often rely on a powerful computer hardware platform, which generally requires multiple GPUs to accelerate. On the other hand, the computing resources are limited. When the user terminal or the mobile terminal uses the above-mentioned deep learning-based human body posture detection algorithm for video data, the computing power of the terminal often cannot meet the demand, so how to reduce or transfer the computational complexity of the human body posture estimation algorithm is also an Important research direction in the field. For a large amount of existing video data, most video analysis tasks are performed by directly identifying the network to all frames of the video[5]. These methods consume a lot of computing resources without considering the time correlation between video frames.

There is a strong spatiotemporal correlation between adjacent frames of video [6] . This correlation is determined by the continuity of motion. Therefore, the moving target and human pose information in the adjacent frames of the video have stronger spatiotemporal correlation. It can be seen that the human body posture information between the video frames has a strong time correlation, and the correlation between the video frames and the motion information can be fully utilized to propagate the detected human body posture information to adjacent frames with higher correlation, thereby Avoid complex human posture detection for each frame of image.

Therefore, a fast 3D human pose estimation algorithm based on optical flow and particle filtering. At the same time is proposed in this paper, we build a system called 3DPLab based on this algorithm. In general, 3DPLab system is mainly composed of three parts, which are key frame detection, multi-joint trajectory tracking and 3D human pose estimation. Firstly, the first frame of the video is used as the key frame. Then, by the method of fast optical flow, it is judged whether the next frame is a key frame. Our method uses 3D human pose estimation for key frames and particle filtering for non-key frames. Since the discrimination between frames is improved, the selection of key frames greatly reduces the time complexity of the algorithm. The efficient particle filter tracking algorithm also greatly improves the speed of 3D human pose estimation while maintaining the recognition accuracy.

In summary, we make the two following contributions:

1.We return to the actual situation of human motion and extracted key frames from the video.

2.We have achieved efficient prediction of 3D human pose estimation in deep learning.

## 2. Pose Estimation and 3D Projection

We have developed a bottom-up approach for fast 3D human pose estimation. Each frame in the video is input to the 3DPLab system, and our system distinguishes between key and non-key frames based on optical flow. Keyframes are motion-sense frames and vice versa. Subsequently, the key frame is the input of the 3D human pose estimation algorithm, and the non-key frame is the input of the multi-joint trajectory tracking based on multi-subspace particle filter algorithm.

### 2.1. Key Frame Detection Based on Optical Flow

The Flownet2-c algorithm used in this paper can obtain optical flow information between key frames and non-key frames [7]. Compared with other optical flow algorithms, the Flownet2-c algorithm maintains a lower error rate while ensuring the efficiency of the algorithm. Optical flow information will be applied by our method to propagate keyframe pose information to non-keyframes. When there is a large displacement between the key joints in the same video frame group and the same joint point in the non-key frame, the

optical flow information cannot accurately describe the motion of the joint point, thereby causing the non-key frame human pose prediction failure.

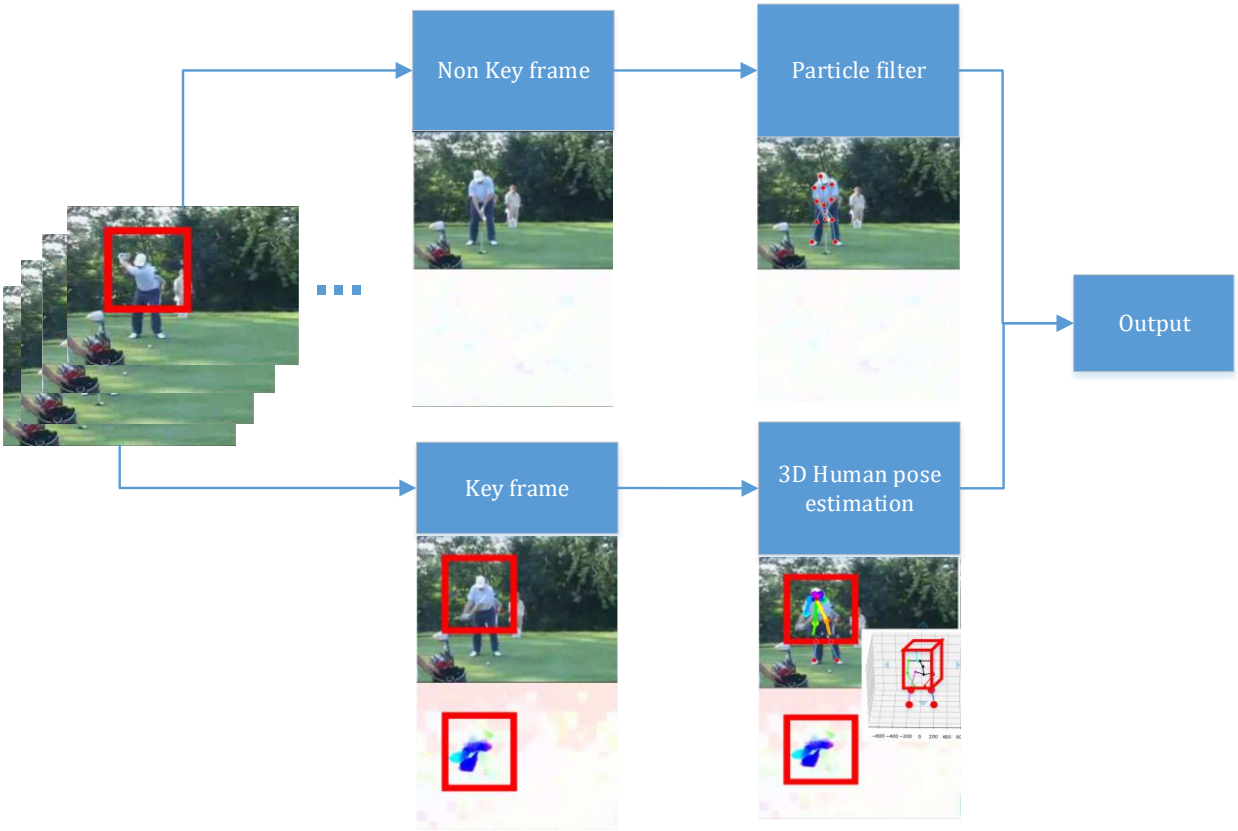


Fig. 1. 3DPLab system framework.

Fig. 2. Key frames detected by our key frame detecting framework. The first row shows key frame are very sparse among the whole video, and the second row shows that key frames and non-key frames (blank frames mean non-key frames). Note that frames are sampled with fixed time interval.

Our method estimates the human body pose of the key frame through the human body posture estimation stage, and obtains the human joint point of the key frame. Then the dense optical flow is used to predict how the joint point should flow to the next frame in time [8].The optical flow algorithm we use can obtain optical flow information between key and non-keyframes. But the shadows or noise in the video are especially noticeable around the moving object, as shown in Fig. 2 If the optical flow information is calculated by the optical flow method. It can be seen from Fig. 2 that the optical flow information around the moving object is very uneven. Therefore, if the key frame attitude information and optical flow information are fused, only the optical flow information at the joint of the key frame is used as the motion information of the joint point of the non-key frame. In this case, the prediction of joint point information

fails due to inaccurate calculation of optical flow information. In this paper, the neighborhood characteristics are used to determine the motion vector of the joint point based on the optical information of the neighboring pixel. The optical information of the $7\times7$ neighborhood at the joint point is used to replace the motion information of the joint point to improve the accuracy of the fusion prediction. The optical flow algorithm uses the following equation (1) to calculate the joint points of non-key frames.

$$\begin{cases} Kf(x_i, y_i) = \dfrac{1}{49}\sum_{n=-3}^{3}\sum_{m=-3}^{3} f(x_i+n, y_i+m) \\ D'(x_i, y_i) = add(D(x_K, y_K) + Kf(x_i, y_i)) \end{cases} \tag{1}$$

where $Kf(x_i, y_i)$ is the mean value of the optical flow information of the $7\times7$ neighborhood at the joint point of the key frame, $D(x_K, y_K)$ is the key frame joint point coordinate, $D'(x_i, y_i)$ is the non-key frame joint point coordinate.

## 2.2. Multi-joint Trajectory Tracking Based on Multi-subspace Particle Filter

Particle filtering is a Bayesian continuous estimation filtering method that achieves the goal tracking by estimating the posterior probability distribution of the state by a weighted particle set [9]. In this section we apply particle filtering to human multi-joint trajectory tracking.

For each predicted 3D joint, we create a corresponding improved particle filter for it. The initial position of the particle filter is the position of the first frame, and the subsequent frames will be predicted, for adjacent reasonable frames. After the content, we will stop the estimation and re-estimate the attitude. This way, the time efficiency of the algorithm is greatly improved, and the real-time performance of the whole framework is improved, which has great practical significance. Therefore, we use a Multi-joint tracking based on multi-subspace particle filter algorithm.

In the Multi-joint tracking algorithm, the new joint detection is divided into four steps: sampling, observation, sample clustering and new joint initialization. First, the sample is not sampled in the independent region contained in the subspace of any joint. The number of samples used for sampling is fixed. It is represented by $N$. At time *t*, the $N$ particles are collected from $Y$ to obtain the $\left\{s_t^{i:0}, i = 1, \ldots, N\right\}$. After the sampling is completed, the weight of the particles in the set is calculated by the observation step, and a threshold $\eta$ is specified, and the sample whose weight is greater than $\eta$ is a valid sample. In order to reduce the amount of calculation, the valid samples obtained by sampling are retained, and other samples are discarded. Since the Multi-joint tracking based on multi-subspace particle filter algorithm adopts a strategy of independently tracking each joint. Therefore, distance-based clustering is performed on the samples, and samples belonging to the same target are grouped into the same sample set. The number of sample sets is the new target number $M_t^N$ at time *t*.

The particle weight is normalized within each sample set. For the new joint *i*, the offset matrix is initialized to *0*, and the noise maximum estimate $e_t^i = e^*$ is obtained, and the state estimate $\overset{i}{x}_{t+1} = x_t^i$ at the next moment is obtained. The subspace $C_{t+1}^i = \{x_t^i, \zeta e^*\}$ is assigned to it according to the subspace definition.

Perform the above operation $M_t^N$ times, and finally get the subspace set $\{C_{t+1}^i, i = 1, 2, \ldots, M_t^N\}$ of the new joint.

The joint tracking is performed in multiple channels, and the subspace of the joint in each tracking represents a tracking channel, and has a sample set independently. The observation step is first performed,

and at time $tt$, the weight of the particles in the sample set $\{s_t^j, i=1,2,...,N_t^i\}$ is calculated for the joint $i$ and its subspace $C_t^i$. Count the number of particles $U_t^i$ with weights higher than $\eta$ in the particle set. If $U_t^i = 0$, the joint fails to track and stop tracking; otherwise, all samples in the sample set are weighted and normalized to obtain a posterior in the target subspace. Probability distribution $P(x_t \mid C_t^i) = \sum_{j=1}^{N_t^i} \pi_t^{j:N_t^i} s_t^{j:N_t^i}$. The state of the joint subspace is then updated by the prediction step. The offset matrix $A_t^i$ is calculated according to equation (2), thereby obtaining the joint state estimation value $\bar{x}_{t+1}$ at the next moment.

$$A_t^i = \| x_t^i - x_{t-1}^i \| \tag{2}$$

The noise maximum value $e_t^i$ is updated according to the update equation (3), that is, the subspace $C_{t+1}^i$ of the next time target is obtained. Repeat the above steps to process all joints and get the number of joints $M_t^D$ that failed to track.

$$e_t^i = \begin{cases} e^*, \text{if } i = 1 \\ \zeta_t e_{t-1} + (1-\zeta_t)\max(v_t^{i:x} - v_{t-1}^{i:x}, v_t^{i:y} - v_{t-1}^{i:y}), \text{if } I_i > 0 \end{cases} \tag{3}$$

After the above operation is completed, re-sampling is entered. For joint $i$, global uniform random sampling is performed within $C_{t+1}^i$ to obtain a new sample set, and the number of samples $N_{t+1}^i$ is calculated according to equation (4).

$$N_t^i = \frac{\pi(\gamma e_{t-1}^i)^2}{S} N^* \tag{4}$$

Repeat the above operation to finally get the set $\{C_{t+1}^i, (s_{t+1}^j), i=1,2,...,M_{t+1}; j=1,2,...,N_{t+1}^i\}$.

## 2.3. 3D Human Pose Estimation

One of the fundamental challenges in creating a human pose model is that it is not possible to obtain sufficiently diverse 3D data to describe the spatial characteristics of the human pose. To compensate for the lack of data, we identified and eliminated confounding factors such as ground plane rotation, limb length, and bilateral symmetry, which led to the inability to identify conceptually similar poses in the training data. We eliminated some factors by simple preprocessing. By normalizing the data, the sum of the squares of the limb lengths on the human skeleton is 1, thus solving the difference in size; while the left and right symmetry is to flip each pose on the x-axis and re-annotating the left to the right, and vice versa.

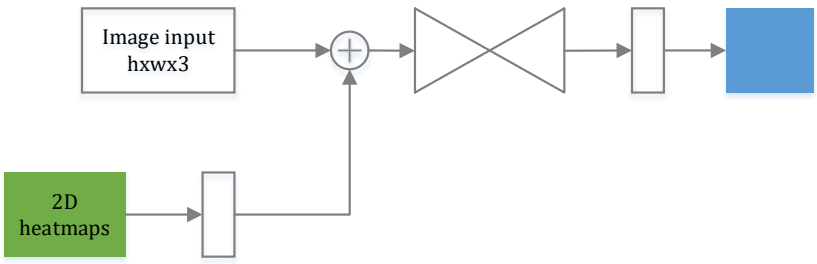


Fig. 3. The blue squares represent the 3D heat map and the green squares represent the 2D pose heat map. The hourglass module is ConvNet. Rough to fine architecture: We use a 2D pose heat map as an intermediate monitor, and then combine it with image features to effectively pass information from the 2D position of the image and joints and joint links.

### 2.3.1. Pose estimation using PAFs

Given a set of detected body parts, how do we assemble them to form the full-body poses of an unknown number of people? We need a confidence measure of the association for each pair of body part detections, i.e., that they belong to the same person. PAFs preserves both location and orientation information across the region of support of the limb[1]. The part affinity is a 2D vector field for each limb. For each pixel in the area belonging to a particular limb, a 2D vector encodes the direction that points from one part of the limb to the other. Each type of limb has a corresponding affinity field joining its two associated body parts.

First, the original input video will be truncated. These intercepted frame sequences are used as input to the 2D human pose estimation algorithm. For 2D pose estimation, we will use a bottom-up approach. The method first detects all the joint points of the human body in the picture, and then uses the global greedy method for all the joint points of the human body to connect the joint points to generate our two-dimensional heat map. Fig. 4 shows the training method of the network.

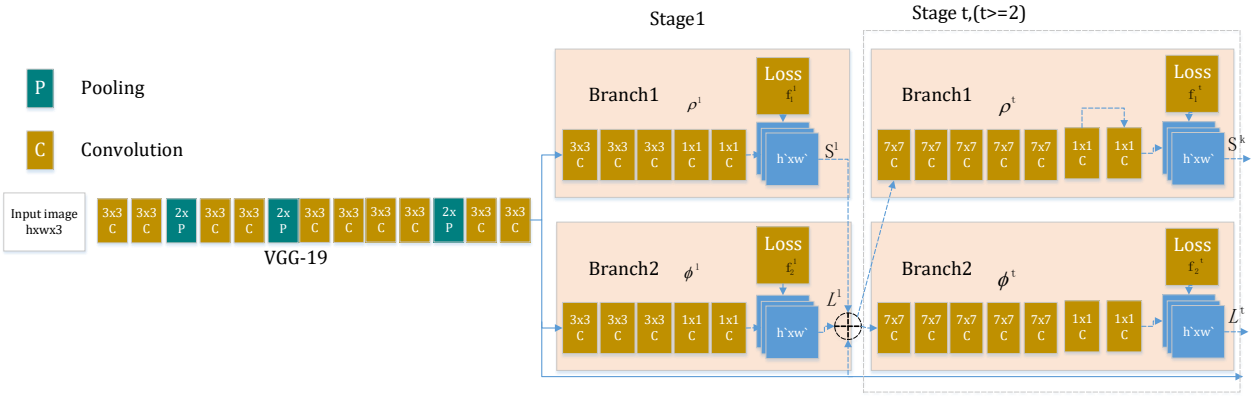


Fig. 4. Architecture of the two-branch multi-stage CNN. Each stage in the first branch predicts confidence maps $S^t$, and each stage in the second branch predicts PAFs $L^t$. After each stage, the predictions from the two branches, along with the image features, are concatenated for next stage.

The body contour estimation system based on partial affinity from coarse to fine is processed by a convolutional neural network on the possible target detection, and the feature atlas F is generated, and the feature atlas F generated after processing is sent as an input to the first stage. It can be seen that the input of the first stage is only related to F, and the input of the stage t is related to three variables. Then through this iteration of multiple stages, we get our 2D joint pose, then predict the possibility of each joint voxel by finely discretizing the three-dimensional space around the theme and training a ConvNet. Training details and equations will not be specifically pointed out in this article.

### 2.3.2. Coarse-to-fine 3D mesh projection

In order to improve the learning ability, a volumetric representation of a 3D human posture is utilized. The volume around the object is evenly dispersed in each dimension. Create a volume size $w \times h \times d$ for each joint. Let $p^n_{(i,j,k)}$ be the predicted possibility of joint $n$ in voxel $(i, j, k)$. To train the network, supervision is also provided in volume. The goal of each joint is in the 3D grid, where there is a three-dimensional Gaussian distribution around the $x^n_{gt} = (x, y, z)$ of the true position of the joint, and the probability that a joint true value coordinate $(x, y, z)$ falls into the voxel of the $(i, j, k)$ volume is defined as:

$$G_{i,j,k}(x_{gt}^n) = \frac{1}{2\pi\sigma^2} e^{\frac{(x-i)^2+(y-j)^2+(z-k)^2}{2\sigma^2}} \tag{5}$$

The error function is defined as follows:

$$L = \sum_{n=1}^{N} \sum_{i,j,k} \left\| G_{(i,j,k)}(x_{gt}^n) - p_{(i,j,k)}^n \right\|^2 \tag{6}$$

The above problem is defined in a way that simplifies the solution of the problem. It also provides a good foundation for the later Coarse-to-fine.

One of the main advantages of volume representation is that it converts highly nonlinear direct 3D coordinate regression problems into more manageable prediction forms in discrete space. In this case, the prediction does not necessarily guarantee a unique position for each joint, but rather provides a confidence estimate for each voxel. This makes it easier for the network to learn the target mapping. A similar point was also raised in the 2D pose case, which validated the benefits of predicting the probability of each pixel instead of pixel coordinates. In terms of network architecture, an important benefit of capacity representation is that it supports predictions using a full convolutional network. Here, we use the hourglass design. This results in fewer network parameters than coordinate regression or pose classification using a fully connected layer. Finally, in terms of predictive output, in addition to being more accurate, our network predictions in the form of dense 3D heat maps are also useful for subsequent post-processing applications. For example, structural constraints can be achieved by using a 3D graphical structure model. Use intensive prediction in a filtering framework while multiple input frames are available.

In this case, the prediction does not necessarily promise a unique position for each joint, but rather provides an estimate of the confidence for each voxel. This makes it easier for the network to learn the target mapping. A similar argument has been previously proposed in the 2D pose case, verifying the benefits of predicting per pixel but not pixel coordinates [2], [10]. An important benefit of volume representation in terms of network architecture is its ability to predict using a full convolutional network.

For each predicted 3D joint, we create a corresponding improved particle filter for it. The initial position of the particle filter is the position of the first frame, and the subsequent frames will be predicted, for adjacent reasonable frames. After the content, we will stop the estimation and re-estimate the attitude. This way, the time efficiency of the algorithm is greatly improved, and the real-time performance of the whole framework is improved, which has great practical significance.

## 3. Experimental Evaluation

The experiments were performed on a desktop with an Intel i7 3.4G CPU, 60G RAM and a Tesla k80 GPU.

We present extensive quantitative evaluation of our coarse-to-fine volumetric approach on three standard benchmarks for 3D human pose: Human3.6M [11], HumanEva-I [12] and KTH Football II [13]. Additionally, qualitative results are presented on the MPII human pose dataset [14], since no 3D groundtruth is available.

Human3.6M dataset: The model was trained and tested on the Human3.6M dataset consisting of 3.6 million accurate 3D human poses [13]. This is a video and mocap dataset of 5 female and 6 male subjects, captured from 4 different viewpoints, that show them performing typical activities (talking on the phone, walking, greeting, eating, etc.).

To evaluate the components of our approach, we use Human 3.6M to report the results as it is the most complete available benchmark. Volume representation: Our first goal is to prove that regression in discrete space is more beneficial than coordinate regression. Both versions of the implementation use the simplest

hourglass settings. The only difference between the two architectures is that the network used for capacity prediction is fully convolved, and for coordinate regression, we end up using a fully connected layer. The results are shown in Table 1. The coordinate regression error of 112.41 mm is similar to the recently reported coordinate regression target output [15]-[18]. In contrast, by using a volumetric output target, a significant reduction in error can be observed, dropping to 85.82 mm at the highest depth resolution.

Comparison with state-of-the-art

Human3.6M: We compared the performance of our method with the previously reported results of human 3.6M. Table 1 gives the average of the 3D error for each joint. Note that some of the previous work [19-21] used a series of frames for pose prediction, not the single frame we considered.

3D Evaluation: Several evaluation protocols have been followed by different authors to measure the performance of their 3D pose estimation methods on the Human3.6M dataset. Tables 1 and 2 show comparisons of the 3D pose estimation with previous works, where we take care to evaluate using the appropriate protocol.

Table 1. Quantitative Comparison on Human3.6M

| | Directions | Discussion | Eating | Greeting | Phoning | Photo | Posing | Purchases |
|---|---|---|---|---|---|---|---|---|
| LinkDE[11] | 132.71 | 183.55 | 132.37 | 164.39 | 162.12 | 205.94 | 150.61 | 171.31 |
| Li *et al.*[22] | - | 134.13 | 97.37 | 122.33 | - | 166.15 | - | - |
| Tekin *et al.*[19] | 102.41 | 147.72 | 88.83 | 125.28 | 118.02 | 182.73 | 112.38 | 129.17 |
| Zhou *et al.*[20] | 87.36 | 109.31 | 87.05 | 103.16 | 116.18 | 143.32 | 106.88 | 99.78 |
| Martinez *et al.*[23] | 65.7 | 68.8 | 92.6 | 79.9 | 84.5 | 100.4 | 72.3 | 88.2 |
| Tome *et al.*[24] | 64.98 | 73.47 | 76.82 | 86.43 | 86.28 | 110.67 | 68.93 | 74.79 |
| Fang *et al.* [25] | 57.5 | 57.8 | 81.6 | 68.8 | 75.1 | 85.8 | 61.6 | 70.4 |
| **Ours** | 67.38 | 71.95 | 66.7 | 69.06 | 71.95 | 76.97 | 65.03 | 68.3 |
| **Ours (with particle filter)** | 70.21 | 72.37 | 69.8 | 73.21 | 73.42 | 77.3 | 67.61 | 68.57 |

| | Sitting | Sitting Down | Smoking | Waiting | Walk Dog | Walking | Walk Together | Average |
|---|---|---|---|---|---|---|---|---|
| LinkDE[11] | 151.57 | 243.03 | 162.14 | 170.69 | 177.13 | 96.60 | 127.88 | 162.14 |
| Li *et al.* [22] | - | - | - | - | 134.13 | 68.51 | - | - |
| Tekin *et al.*[19] | 138.89 | 224.9 | 118.42 | 138.75 | 126.29 | 55.07 | 65.76 | 124.97 |
| Zhou *et al.*[20] | 124.52 | 199.23 | 107.42 | 118.09 | 114.23 | 79.39 | 97.7 | 113.01 |
| Martinez *et al.* [23] | 109.5 | 130.8 | 76.9 | 81.4 | 85.5 | 69.1 | 68.2 | 84.9 |
| Tome *et al.*[24] | 110.19 | 173.91 | 84.95 | 85.78 | 86.26 | 71.36 | 73.14 | 88.39 |
| Fang *et al.* [25] | 95.8 | 106.9 | 68.5 | 70.4 | 73.89 | 58.5 | 59.6 | 72.8 |
| **Ours** | 83.66 | 96.51 | 71.74 | 65.83 | 74.89 | 59.11 | 63.24 | 71.9 |
| **Ours (with particle filter)** | 83.9 | 99.14 | 73.38 | 68.75 | 79.26 | 62.42 | 66.52 | 73.86 |

Table 2. Valuation of Particle Filter and 3D Pose Estimation for Single Person Video on Human3.6M.

| | *fps* |
|---|---|
| LinkDE [11] | 10 |
| Li *et al.* [22] | 16 |
| Tekin *et al.* [19] | 13 |
| Zhou *et al.* [20] | 12 |
| Martinez *et al.* [23] | 19 |
| Tome *et al.* [24] | 12 |
| Fang *et al.* [25] | 14 |
| Our 3D pose estimation with Particle filter | 48 |
| Our 3D pose estimation without Particle filter | 27 |

The numbers are the average 3D joint error (mm). Baseline numbers are taken from the respective papers. Note, several approaches use video for prediction rather than a single frame[20].

This paper uses the number of frames processed per second to evaluate the algorithm detection speed.

$$Fps = \frac{nFrame}{\sum_{i=1}^{nFrame} t_i} \tag{7}$$

where Fps is the number of frames processed per frame, *nFrame* is the number of frames of the test video, and $t_i$ is the detection time of the i*th* frame, which includes calculating the optical flow information between the two pictures by the optical flow method for 10 ms.

For Particle filter, faster estimation is better. For pose estimation evaluation, lower error is better.

Table 3. Evaluation of Particle Filter and 3D Pose Estimation for Single Person Image on Human3.6M

| | *millisecond* |
|---|---|
| LinkDE [11] | 100 |
| Li *et al.* [22] | 67 |
| Tekin *et al.* [19] | 77 |
| Zhou *et al.* [20] | 83 |
| Martinez *et al.* [23] | 53 |
| Tome *et al.* [24] | 73 |
| Fang *et al.* [25] | 71 |
| Our 3D pose estimation | 37 |

Table 4. Evaluation of Particle Filter and 3D Pose Estimation for 2 Persons' Video on UCF-101

| | *fps* |
|---|---|
| LinkDE [11] | - |
| Li *et al.* [22] | - |
| Tekin *et al.* [19] | - |
| Zhou *et al.* [20] | - |
| Martinez *et al.* [23] | - |
| Tome *et al.* [24] | - |
| Fang *et al.* [25] | - |
| Our 3D pose estimation with Particle filter | 38 |
| Our 3D pose estimation without Particle filter | 17 |

In order to detect the efficiency of the algorithm, we use an intuitive and effective detection method, which measures the efficiency of the entire 3D human pose estimation system by *fps*. Firstly, the algorithm runtime of single-person 3D pose estimation is used to measure the runtime of the open source system involved (Table 3). To evaluate the speed of single-person pose estimation in each algorithm, we evaluated some of the action videos in the Human3.6m dataset.

Next in importance, many 3D human pose estimation algorithms do not support multi-person detection. Therefore, in order to evaluate the superiority of our method, the 3D human pose estimation algorithm proposed in this paper is embedded in our system for evaluation. This paper adopts a more realistic human body 3D attitude estimation evaluation method, which is to evaluate the running efficiency of the algorithm in multiple video sequences, and to present the algorithm efficiency intuitively and realistically.

Thirdly, to evaluate the generalization and robustness of the entire 3D human pose estimation system, we used an extended field data set, UCF-101. The data set has a video sequence time of approximately 10 s and each video sequence contains approximately 150 frames. We evaluated in a video sequence of multiple action types (Table 4).

The above results show that the proposed acceleration algorithm can improve the processing speed and reduce the computational complexity by using the time correlation between video frames and the multi-joint particle filter tracking when the average detection accuracy is improved. The experimental data indicates that the selection of key frames by the key frame detection algorithm of the system reduces the time complexity of human pose estimation, and the speed of 3D human pose estimation is improved by the application of the efficient multi-joint trajectory tracking algorithm while maintaining the estimation

accuracy.

## 4. Conclusion

In this paper, we propose a fast human pose estimation algorithm based on optical flow and particle filter. We evaluate our proposed framework on Human3.6M dataset and achieve significant improvement on the time efficiency of 3D human pose estimation. We also demonstrate that the computational complexity of the fast 3D pose estimation algorithm proposed in this paper is lower than that of the human pose estimation algorithm. In this paper, the computational complexity of human pose estimation can be effectively reduced and the detection speed can be improved when the detection effect is not much different from the original algorithm. In future, we should further consider how to efficiently select key frames and further accelerate the algorithm.

## Acknowledgment

This work was sponsored by National Key R&D Plan of China (grant number 2016YFC0800506 and grant number 2017YFB1201203), National Nature Science Foundation of China (grant number 61702111), National Natural Science Foundation of China (grant number 83-Y40G33-9001-18/20), Opening Project of Guangdong Province Key Laboratory of Cyber-Physical System.

## References

[1] Cao, Z., Simon, T., Wei, S., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*.

[2] Pfister, T., Charles, J., & Zisserman, A. (2015). Flowing ConvNets for human pose estimation in videos. *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*.

[3] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2018). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[4] Pavlakos, G., Zhou, X., Derpanis, K. G., & Daniilidis, K. (2016). Coarse-to-fine volumetric prediction for single-image 3D human pose. Retrieved from: https://ui.adsabs.harvard.edu/\#abs/2016arXiv161107828P

[5] Han, S., Mao, H., & Dally. W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. Retrieved from: https://ui.adsabs.harvard.edu/\#abs/2015arXiv151000149H

[6] Kang, D., Emmons, J., Abuzaid, F., Bailis, P., & Zaharia. M. (2017). NoScope: Optimizing neural network queries over video at scale. Retrieved from: https://ui.adsabs.harvard.edu/\#abs/2017arXiv170302529K

[7] Ilg, E., N., Mayer, T., Saikia, M., Keuper, A., Dosovitskiy, & Brox, T. (2017). FlowNet 2.0: Evolution of optical flow estimation with deep networks. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Zuffi, S., Romero, J., Schmid, C., & Black, M. J. (2013). Estimating human pose with flowing puppets. *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia.

[9] Isard, M., & Blake, A. (1998). CONDENSATION — Conditional density propagation for visual tracking. *International Journal of Computer Vision, 29*, 5-28.

[10] Tompson, J., Jain, A., Cun, Y. L., & Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Computer Science*.

[11] Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Trans Pattern Anal Mach Intell,*

*36*, 1325-1339.

[12] Sigal, L., Balan, A. O., & Black, M. J. (2009). HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision, 87*.

[13] Kazemi, V., Burenius, M., Azizpour, H., & Sullivan, J. (2013). Multi-view body part recognition with random forests. *Proceedings of the 2013 24th British Machine Vision Conference*.

[14] Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[15] Li, S., & Chan, A. B. (2014). 3d human pose estimation from monocular images with deep convolutional neural network. *Proceedings of the Asian Conference on Computer Vision*.

[16] Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., & Fua, P. (2016). Structured prediction of 3d human pose with deep neural networks.

[17] Park, S., Hwang, J., & Kwak, N. (2016). 3D human pose estimation using convolutional neural networks with 2D pose information. *European Conference on Computer Vision*.

[18] Zhou, X., Sun, X., Zhang, W., Liang, S., & Wei, Y. (2016). Deep kinematic pose regression. *Proceedings of the European Conference on Computer Vision*.

[19] Tekin, B., Rozantsev, A., Lepetit, V., & Fua, P. (2015). Direct prediction of 3D body poses from motion compensated sequences. Retrieved from: https://ui.adsabs.harvard.edu/\#abs/2015arXiv151106692T

[20] Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., & Daniilidis, K. (2016). Sparseness meets deepness: 3D human pose estimation from monocular video. *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*.

[21] Du, Y., Wong, Y., Liu, Y., Han, F., Gui, Y., Wang, Z., *et al.* (2016). Marker-less 3d human motion capture with monocular image sequence and height-maps. *Proceedings of the European Conference on Computer Vision*.

[22] Li, S., Zhang, W., & Chan, A. B. (2015). Maximum-margin structured learning with deep networks for 3D human pose estimation. Retrieved from: https://ui.adsabs.harvard.edu/\#abs/2015arXiv150806708L

[23] Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A simple yet effective baseline for 3d human pose estimation. *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*.

[24] Tome, D., Russell, C., & Agapito, L. (2017). Lifting from the Deep: Convolutional 3D pose estimation from a single image. *IEEE Conference on Computer Vision & Pattern Recognition*.

[25] Fang, H., Xu, Y., Wang, W., Liu, X., & Zhu, S.-C. (2018). Learning pose grammar to encode human body configuration for 3D pose estimation.

**Jiaying Lan** was born in Jiangmen, Guangdong, China in 1998. He is majoring in software engineering at Guangdong University of Technology, Guangzhou, China. His research interests include human pose estimation, and deep learning.

**Guoheng Huang** is currently an IEEE member, a CCF member and a talented person in the "hundred talents program" of Guangdong University of Technology, an assistant professor of computer science, and a master's tutor. He received his bachelor of science (mathematics and applied mathematics) and master of engineering (computer science) degrees from

South China Normal University in 2008 and 2012, and his Ph.D. degree (software engineering) in Macau University in 2017. His research interests include computer vision, pattern recognition and artificial intelligence. He has hosted and undertaken a number of national and provincial-level scientific research projects, including the natural science foundation of China and National Key Research and Development Plan etc. He has published many research papers and been as a key member of Guangdong Key Laboratory of Cyber-Physical System.

**Lianglun Cheng** received the B.E. and M.S. degrees in automation from Huazhong University of Science and Technology, Wuhan, China, in 1988 and 1992, respectively. He received the Ph.D. degree in automation from Chinese Academy of Sciences in 1999.

Since 2003, he has been a professor at Guangdong University of Technology, Guangzhou, China. He is currently the dean of the School of Computer, Guangdong University of Technology. His research interests include terahertz detection technology, visual image processing and cyber-physical systems.