A MARKERLESS BODY MOTION CAPTURE SYSTEM FOR CHARACTER ANIMATION BASED ON MULTI-VIEW CAMERAS

Jinbao Wang, Ke Lyu, Jian Xue, Pengcheng Gao, Yanfu Yan*

University of Chinese Academy of Sciences, School of Engineering Science, Beijing

ABSTRACT

A novel application system is proposed in this paper to achieve the generation of 3D character animation driven by markerless human body motion capture. The whole pipeline of the system consists of four parts: capturing motion data by multiple cameras, detecting 2D human body joints and estimating 3D joints, calculating bone transformation matrices, and generating character animation. Its main objective is to generate 3D skeleton and animation for 3D characters from multi-view images captured by ordinary cameras. The computation complexity of 3D skeleton reconstruction based on 3D vision is reduced accordingly to achieve the frame-by-frame motion capture. The experimental results show that our system is effective and efficient for capturing human action and animating 3D cartoon characters simultaneously.

Index Terms— 3D Vision, Markerless Motion Capture, Multi-View Cameras, Character Animation, Convolutional Neural Networks.

1. INTRODUCTION

The study of human body motion capture has made considerable progress over recent decades, which is driven by the practical requirements in the applications of entertainment, sports and clinical, etc. Most previous work requires high quality cameras or a chromatic background to segment the person in foreground precisely. Some recent methods utilize depth sensors to improve the efficiency of data acquisition. However, these methods aim at the body shape or skeleton reconstruction, which are usually expensive and not suitable for normal applications.

To satisfy the requirements of some applications (e.g. 3D animation, VR games) for simple and easy-to-use motion capture techniques, this paper presents a markerless body motion capture system based on multi-view cameras. This system can use few cheap cameras to generate the body skeleton and animate characters effectively by 4 stages.

The first stage is motion data capture with multiple cameras, including camera calibration and synchronization. After capturing multi-view images, CNN pose detector is taken to detect the 2D skeleton. Next, the second stage is the calculation of the 3D skeleton based on multi-view geometry constraint. In the third stage, bone transformation is calculated using 3D skeleton to prepare for the last stage. At last, the character rigging is achieved based on 3D skeleton gotten from last stage and optimized in sequence to animate 3D character vividly.

2. RELATED WORK

Motion capture is the process of recording the movement of objects or people. A large variety of systems and approaches have been proposed for capturing human motion in the past years. There are two categories as follows.

Marker-based motion capture. Currently, motion capture based on markers has been a mature technique successfully used in many fields, i.e. the movie industry and virtual reality. While this method is usually suffering from inconvenience that controller need to wear marker suits with sensors, such as optical markers [1], or mounted cameras [2], making them unable to capture motions of people wearing everyday clothing. In addition, marker based motion capture is sensitive to skin movement relative to the underlying bone [3].

Markerless motion capture. Recently, more and more researchers pursue this hot topic. From an algorithmic point of view, markerless motion capture can be classified into two main categories: discriminative approaches [4] and generative approaches [5]. Discriminative approaches take advantage of data driven machine learning strategies to convert the motion capture problem into a regression or pose classification problem [6, 7], and therefore are suitable for human-computer interaction applications where efficiency is more important than accuracy. As for generative approaches for motion capture, the ultimate goal is to acquire the pose and shape of the body, which is achieved by fitting the model to information extracted from the images. These methods can generate a set of model parameters such as body shape, bone lengths and joint angles. In contrast of discriminative approaches, generative approaches are usually based on temporal information and solve a tracking problem. The motion capture process

*The work is supported by National Key R&D Program of China under contract No. 2017YFB1002203, National Natural Science Foundation of China (NSFC, Grant No. 61671426, 61731022, 61471150, 61572077), the Beijing Natural Science Foundation (Grant No. 4182071), the Instrument Developing Project of the Chinese Academy of Sciences (Grant No. YZ201670). (*Corresponding auther: Jian Xue; email: xuejian@ucas.ac.cn*)

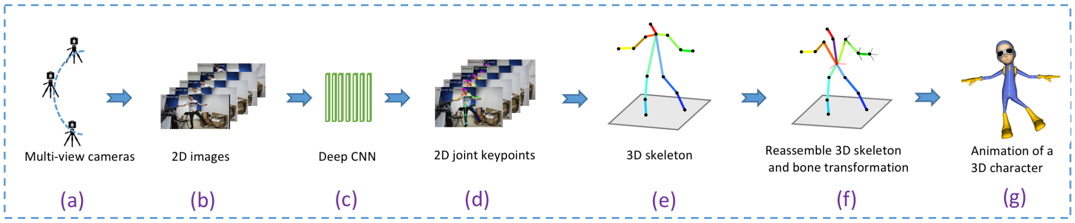

Fig. 1. The processing pipeline. (a) multi-view cameras, (b) images captured, (c) 2D joint keypoints detector, (d) output of 2D joint keypoints, (e) 3D skeletion, (f) reassembled 3D skeleton and bone transformation, (g) animation of a 3D character.

is formulated as a frame-by-frame optimization to deform the skeletal pose [8], the surface geometry [9, 10] or both [11, 12].

Relation to prior work. *Earlier, Baran et al.* [13] presented a method for animating characters automatically, which adopts the skeleton to the character and attaches it to the surface, allowing skeletal motion data to animate the character. Film *et al.* [14] proposed an open source system called OpenMoCop for optical motion capture, which is developed based on digital image analysis techniques. Recently, there are many methods to obtain the human body joints or shape, by using human 3D scanning or reconstruction [12, 10, 15], CNN based method [7, 16], and so on, but there is a lack of complete system to animate virtual avatars or characters. In addition, to get fine whole body by 3D reconstruction is difficult and costs a large amount of computing resources, so it is very hard to apply in the commercial area. Besides, it is a challenge to use the CNN based method to acquire the 3D joints with good performance. So our contribution is that we propose a entire system that can capture human body motion by no markers easily and fast, concentrating on the body 3D skeleton reconstruction and virtual character animation.

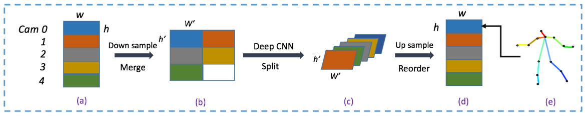

Fig. 2. 2D joint keypoints detection. (a) input 2D images from multi-view cameras, (b) merge these images into one single image, (c) get out-of-order 2D joint keypoints from the Deep CNN detector, (d) reorder the 2D joint keypoints in accordance with camera id, (e) get 2D skeleton for each camera image.

3. APPROACH

The pipeline of our system mainly includes 4 stages, as shown in Figure 1. Stage 1 (a-d): multiple *rgb* cameras shown in (a) are adopted to collect 2D images from multi-views shown in (b). (c) is the 2D human body joints detector to detect the joint keypoints in 2D images, using deep convolutional neural network (Deep CNN). (d) produces the multi-view output images with 2D joints. Stage 2 (d-e): the joint keypoints in 3D space are calculated according to the constraint of multi-view geometry. Stage 3 (e-f): character rigging is achieved by reassembling a new 3D skeleton with bone transformation matrix in (f) according to the skeleton in (e) which only contains the joint keypoints. After above steps, the information obtained for rigging is used to animate a 3D character in Stage 4 (f-g).

3.1. 2D Joints Detection from Multi-view 2D Images

The basic process is shown in Figure 2. First of all, all the cameras are calibrated after their positions are fixed, using fundamental matrix estimation for pairs of images followed by bundle adjustment. Such calibration procedure only needs to be done once for certain layout of cameras. Then, the deep CNN detector in a state-of-the-art open-source 2D pose estimation library [4] is used to perform the detection of the 2D human pose keypoints. While, as known, using the Deep CNN method is time-consuming and costs large amount of computing resources, a procedure of merging multi-view images into one image, which is fed into the deep CNN, is designed to speed up the calculation and improve the synchronization of output 2D poses. And the 2D skeleton id can be obtained with its corresponding camera by reordering.

3.2. 3D joints estimation by multi-view 2D joints

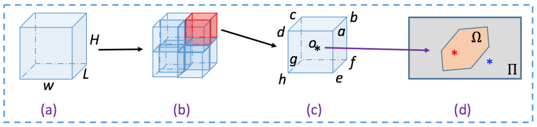

Fig. 3. Estimation of 3D joints by 2D joints. (a) a 3D sample space initialized with width W, height H and length L, (b) space subdivision, (c) one subspace as a 3D sample unit, (d) the projection area Ω of the 3D sample unit on image Π.

An overview of the approach for 3D joints estimation is presented in Figure 3. The basic idea about estimating the 3D joints of our approach is based on feature point correspondence and visual hull method. Compared with other dense reconstruction of complex dynamic scenes from multiple wide-baseline camera views, we just reconstruct sparse 3D points based on matched 2D joint keypoints from the pose detection network. In order to speed up the searching of joint candidates in 3D space, subdivision of the space is performed in our method, which is illustrated in Figure 3.

3.2.1. 3D sample point projection to 2D image area

After initializing one beginning 3D space with W, H, L, the space is split into eight-fold 3D sub-regions, as shown in Figure 3(a and b). Taking a 3D sample point as example shown in Figure 3(c), 9 points (8 vertices and 1 center point) of one sample cube, $\{a, \cdots, g, o\}$, are projected to the image Π captured by i-th camera. The new points after projection by $\{a, \cdots, g\}$ make up an area Ω shown in Figure 3(d).

3.2.2. 3D joint candidates and iterative space subdivision

At the view of the ith camera, the part area of the 2D image Π_i is defined as Ω_i, and a 2D point is p_i^2. we define a 3D sample cube with width w, height h and length l as $Cube_{\{w,h,l\}}$, and its center point with p^3. $\text{project}(p^3, \kappa)$ represents the projection from p^3 to p^2, where κ is the camera parameters. So the projection wide M_i of the 3D sample cube in the ith image is defined as

$$\Omega_i = \{p^2 \mid p^2 = \text{project}(p^3, \kappa),\ p^3 \in Cube\}.$$

Since a cube projection on the 2D plane is convex, its projection area can be calculated easily by its 8 vertices {a, ..., g}. We define the formula ψ_i as follows.

$$\psi_i = \begin{cases} 1 & p_i^2 \in \Omega_i \\ 0 & \text{otherwise.} \end{cases}$$

In this formula, we consider two cases about the relationship between the real 3D joint keypoint p^3 and their candidates. If the p^2 detected in 2D camera images locates in the projection area M of p^3 candidates, we think the cube maybe contain its corresponding p^3. Here, we define $N_{Cube} = \sum_{i=0}^{n} \psi_i$ under multi-views meeting ψ_i.

An iterative space subdivision strategy is used to locate the keypoint of each joint in 3D space. This iterative process will continue until the stop condition, i.e. $N_{Cube} < \delta$ or size of $Cube < \sigma$, is satisfied. In the stop condition, δ is a integer constant, which is usually less than the number of cameras for better error tolerance, i.e. at least δ camera images must meet the condition $p_i^2 \in M_i$. σ is the smallest cube size for space subdivision, which is set to 0.05 in our work. The size of cube is determined by the 2D image resolution and the projection wide in the 2D image. Usually, the projection wide is decided by the error of pose detector. In each iteration, the edge length of the cube for space subdivision is reduced by half.

According to the procedure described above, all the joints of the captured person are processed one by one and their 3D candidates are obtained finally. Due to adoption of subdivision and sparse 3D keypoints reconstruction, the efficiency of the processing is ensured.

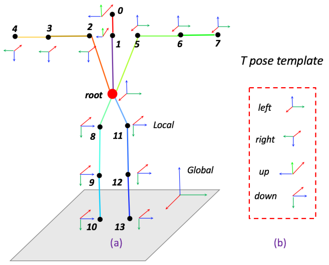

Fig. 4. T pose template: the body's T pose template definition including joints, bones in (a) and local coordinates in (b).

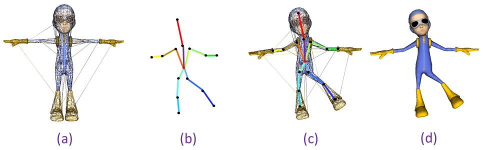

Fig. 5. Character rigging. (a) a character model, *Astroboy*, (b) skeleton setup (rigging), (c) binding skeleton to the model, (d) rendering result.

3.3. Transformation calculation and character rigging

Due to the difficulty of giving absolute transformation matrix for each joint in world coordinate, the transformation matrix relative to the T pose is calculated instead in our method.

The *T pose template* is presented in Figure 4, which defines joints and bones of a 3D skeleton and *left* , *right* , *up* , and *down* local coordinates, letting x axes of local coordinates point to the extending direction of limbs. The transformation matrix T of each joint is defined as

$$T_{4\times4} = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0 & 1 \end{bmatrix}.$$

where R is the 3×3 rotation matrix and t is the 3×1 translation matrix. Here t is obtained from the rigging information of the original character model, and the bone rotation matrix is calculated from the variation between the 3D joints gotten in Section 3.2 and the T pose template. Finally, the process of animation goes like this in Figure 5.

4. EXPERIMENTS

In this section, we present some experimental results of our method and comparison with a commercial optical motion capture system, ASUS Xtion PRO LIVE (Xtion).

Our system is built with a workstation and five same IP cameras (HIKVISION DS-IPC-B12-I). The workstation is

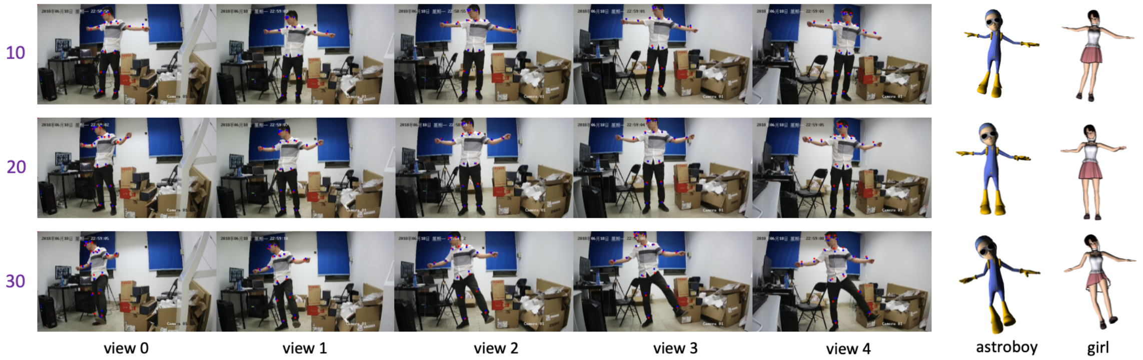

Fig. 6. Comparison of projection joints of 3D voxel and 2d CNN detection joints. 10, 20, 30 are the frame of sequence. It displays different views from view 0 to view 4. Red points are the 2d detection joints by Deep CNN, and blue points are the 2d projection joints of 3D voxel, which is calculated by our method. "astroboy" and "girl" are the cartoon characters.

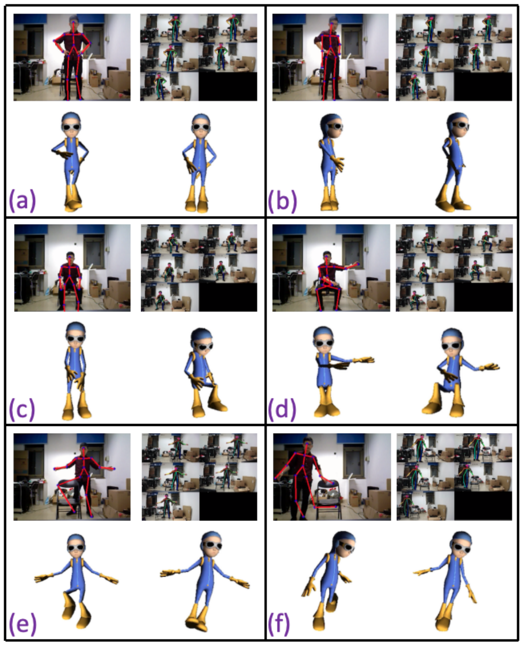

Fig. 7. Failure cases of Xtion with correct results by our system. (a) wrong bone transformation for right upper arm, (b) tracking failure (right upper arm), (c) tracking failure (interfered by other parts), (d) tracking failure (limb overlap), (e) tracking failure (objects in front of limbs), (f) tracking failure (objects very close to limbs).

equipped with a Intel Core i7-6850K@3.6GHz CPU, 32GB RAM and a NVIDIA TITAN X GPU.

Figure 6 shows the comparison of reprojected points of the calculated 3D joint points and the results directly obtained from 2D CNN detector. From the comparison of each view, it shows that our 3D joint points estimation method works well.

Secondly, we compare the performance of our system with the same functionality provided by Xtion, which carries a structure sensor and uses 3D sensing solution provided by PrimeSense company (bought by Apple Inc. in 2013). PrimeSense developed NiTE middleware which analyzed the data from hardware and modules for OpenNI to provide gesture and skeleton tracking.

Compared with Xtion, our solution could get more accurate and robust results, which is indicated by the examples illustrated in Figure 7. Because our system adopts the single-frame processing based method while Xtion utilizes tracking based method, there exists slight jitter in the results produced by our system. However, in some cases Xtion fails to present the skeleton's joints completely corresponding to the person captured by the camera, as shown in Figure 7. There are two main reasons for these failures: tracking error and wrong transformation matrix. When body limbs move quickly or are interfered by limbs itself and other objects, the motion tracking by Xtion always fails, which produces tracking error. Besides, even if tracking is successful, Xtion may give wrong transformation matrices for some joints, which leads to weird action of the animation character, as shown in Figure 7(a).

Due to non-tracking strategy and multi-view configuration, our system can reduce the interference of limbs themselves or other objects and capture the body motion whenever possible. Furthermore, by the precise calculation method of 3D joints, matrix errors rarely occur in our system. Therefore, for most Xtion's failure cases, our system produces correct results, as shown in Figure 7.

5. CONCLUSIONS

In this paper, a new system is proposed for markerless human motion capture and animation character rigging based on multi-view cameras. According to the experimental results, our system can produce accurate and robust 3D human body joints from multi-view camera images, which are used for animation character rigging. This system may be used in the field of animation production, video game production, VR game interaction, etc., which can reduce the production costs and simplify the human-machine interaction notably.

6. REFERENCES

[1] Ramesh Raskar, Hideaki Nii, Bert Dedecker, Yuki Hashimoto, Jay Summet, Dylan Moore, Yong Zhao, Jonathan Westhues, Paul Dietz, and John Barnwell, "Prakash:lighting aware motion capture using photosensing markers and multiplexed illuminators," *Acm Transactions on Graphics*, vol. 26, no. 3, pp. 36, 2007.

[2] Takaaki Shiratori, Leonid Sigal, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins, "Motion capture from body-mounted cameras," in *ACM SIGGRAPH*, 2011, p. 31.

[3] Croce U Della, A Leardini, L Chiari, and A Cappozzo, "Human movement analysis using stereophotogrammetry. part 3: Soft tissue artifact assessment and compensation," *Gait & Posture*, vol. 21, no. 2, pp. 221–225, 2005.

[4] Zhe Cao, Tomas Simon, Shih En Wei, and Yaser Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," pp. 1302–1310, 2016.

[5] J. Duetscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, pp. 126–133 vol.2.

[6] Chaoqun Hong, Jun Yu, Yong Xie, and Xuhui Chen, "Multi-view deep learning for image-based pose recovery," in *IEEE International Conference on Communication Technology*, 2016, pp. 897–902.

[7] Christian and Theobalt, "Vnect: real-time 3d human pose estimation with a single rgb camera," *Acm Transactions on Graphics*, vol. 36, no. 4, pp. 44, 2017.

[8] Carsten Stoll, Nils Hasler, Juergen Gall, Hans Peter Seidel, and Christian Theobalt, "Fast articulated motion tracking using a sums of gaussians body model," in *International Conference on Computer Vision*, 2011, pp. 951–958.

[9] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans Peter Seidel, and Sebastian Thrun, "Performance capture from sparse multi-view video," *Acm Transactions on Graphics*, vol. 27, no. 3, pp. 1–10, 2008.

[10] Kaiwen Guo, Feng Xu, Yangang Wang, and Yebin Liu, "Robust non-rigid motion tracking and surface reconstruction using l0 regularization," in *IEEE International Conference on Computer Vision*, 2015, pp. 3083–3091.

[11] Yebin Liu, C. Stoll, J. Gall, H. P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *Computer Vision and Pattern Recognition*, 2011, pp. 1249–1256.

[12] Y. Liu, J Gall, C Stoll, Q. Dai, H. P. Seidel, and C Theobalt, "Markerless motion capture of multiple characters using multiview image segmentation.," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 11, pp. 2720–2735, 2013.

[13] Ilya Baran, "Automatic rigging and animation of 3d characters," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 72, 2007.

[14] David Lunardi Flam, Daniel Pacheco De Queiroz, Arnaldo De Albuquerque Araújo, and João Victor Boechat Gomide, *OpenMoCap: An Open Source Software for Optical Motion Capture*, IEEE Computer Society, 2009.

[15] L. Xu, Y. Liu, W. Cheng, K. Guo, G. Zhou, Q. Dai, and L. Fang, "Flycap: Markerless motion capture using multiple autonomous flying cameras," *IEEE Transactions on Visualization & Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2016.

[16] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis, "Harvesting multiple views for marker-less 3d human pose annotations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1253–1262.