3D RECONSTRUCTION OF MULTI-VIEW CAMERA AND MARKERLESS TRACKING OF HUMAN POSE AND BASEBALL.

[1]Yen-Ying Lu (呂彥穎), *[1]Yuh-Renn Wu* (吳育任)*, *[2]Chiou-Shann Fuh* (傅楸善)

[1] Graduate Institute of Photonics and Optoelectronics, National Taiwan University, Taipei, Taiwan, 10617

[2] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 10617.

E-mail: r07941023@ntu.edu.tw yrwu@ntu.edu.tw fuh@csie.ntu.edu.tw

ABSTRACT

This paper proposes a model to reconstruct and track the 3D poses of baseball player. The pose of a baseball player is analyzed with AlphaPose, and the baseball position is tracked by brightness, size, shape, and velocity of the ball. Finally, we reconstruct 3D poses after we use the binary search iteration and find the relative orientation between the cameras.

Keywords: *AlphaPose 2D pose estimation, relative orientation, 3D reconstruction.*

1. INTRODUCTION

With the development of technology, high sampling rate cameras becomes more affordable for researchers or even publics. These tools are very important for sport analysis especially for intense excise such as baseball, badminton, or tennis, etc... In this research, we are interested in baseball analysis. When analyzing the performance of player in a baseball game, it is important to focus on watching the path of the ball or the movement of the player's joint point especially in fingers and wrist joint. Therefore, the motion video analysis has been widely used in the broadcast of baseball games. The K-Zone system is already used by the Major League Baseball (MLB), but only analyzes the trajectory of baseball.

Recently, the development of deep learning has made great contributions to our societies. In this paper, we are working on applying the result of deep learning on baseball researches. As we know, not only the performance of athletes can be analyzed with the assist of deep learning, but also the analysis of tactics can be achieved. And even the function of assisting judgment can be achieved. We believe that with a proper model, the occurrence of false positives can be reduced.

One of the purposes in this study is to build the model of tracking baseball players' pitching poses. Traditional, to accurately trace the player's pose, the marker with infrared detector is used to trace the position. However(VICON system[1]), this will affect the player's body movement. The better way is to avoid using the marker with a suitable image recognizer. In recent years, the developments of AlphaPose[2, 3] and YOLOv3[4] and are remarkable for human pose detection. Therefore, we will try to implement these packages to recognize the player's pose. Beside the tracking of player's pose, it is also important to track the movement of baseball in order to understand quality of pitching. Therefore, we will also need to develop the model to trace the trajectory of baseball. Finally, we will need to reconstruct the 3D pose with stereoscopic vision in each image.

2. APPROACH

2.1. Sample

The sample comes from volunteered baseball pitchers, which are students in National Taiwan University of Sports at the ages between 18 and 24. Test samples exclude conditions that will affect routine practice or competition, as well as inability to comply or understand instructions, including shoulder or elbow surgery in the past, musculoskeletal disorders in the limbs in the past 3 months, neuromuscular disorders (such as stroke, spinal cord injury, Peripheral nerve damage, etc.). The tester participated in the study after the researcher's instructions and signed a consent form approved by the Research Ethics Committee of School of Medicine, National Taiwan University.

2.2. Equipment

The filming action consists mainly of three synchronous cameras, the Sony HDR-AS50, with a sampling frequency of 120 Hz and an image resolution of 1280x720 pixels. The spatial geometric distance is mainly measured by an infrared sensor. The calibration image is 12x20 black and white square array with a side length of 10 cm.

The computer operating system is Ubuntu 18.04; the CPU is AMD Ryzen threadripper 1920X, with 12 cores

and 24 threads.; the GPU is GTX 1080 Ti with 12GB memory; and the RAM is 128GB.

2.3. Camera Calibration

A global 3D coordinate is rotated by the camera coordinate R and translated $\vec{T}$, and then is multiplied by the camera Interior Orientation *M*[5] (camera intrinsics matrix) to get $\vec{q}$, and the first two values of $\vec{q}$ constitute the image two-dimensional coordinates, the projection formula is $\vec{q} = M \cdot R\left(\vec{Q} - \vec{T}\right)$, the camera internal matrix is defined as

$$M = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

f_x and f_y are multiplication of the focal length *f* of the lens with the image pixel size s_x, s_y, and then c_x and c_y are the distance at which the center of the imaging wafer is offset from the optical axis.

In fact, it is difficult to perfectly align the lens mechanically. Therefore, we also need to further consider lens distortion, which is divided into radial distortions [6] caused by lens shape and tangential distortions [7, 8] caused by assembly machine error. The pinhole camera model is that some of the light can only pass through the hole, and other light will be blocked. However, the simple model is not suitable for making images because it cannot collect enough light to quickly expose it. Instead, the larger aperture with a large lens is used to try to collect more light to reduce the exposure time, but the shortcomings of this method will cause serious radial distortion in the image, and the correction formula for lens distortion are

Tangential distortions :

$$\begin{cases} x_{corrected} = x + \left[2p_1xy + p_2\left(r^2 + 2x^2\right)\right] \\ y_{corrected} = y + \left[2p_2xy + p_1\left(r^2 + 2x^2\right)\right] \end{cases} \quad (2)$$

Radial distortions:

$$\begin{cases} x_{corrected} = x \cdot \left(1 + k_1r^2 + k_2r^4 + k_3r^4\right) \\ y_{corrected} = y \cdot \left(1 + k_1r^2 + k_2r^4 + k_3r^4\right) \end{cases} \quad (3)$$

First use the OpenCV function[9] to detect the position of the checkerboard in the image in Figure 1, and then solve the camera internal matrix and distortion coefficient, and finally minimize the reprojection error in Figure 2.

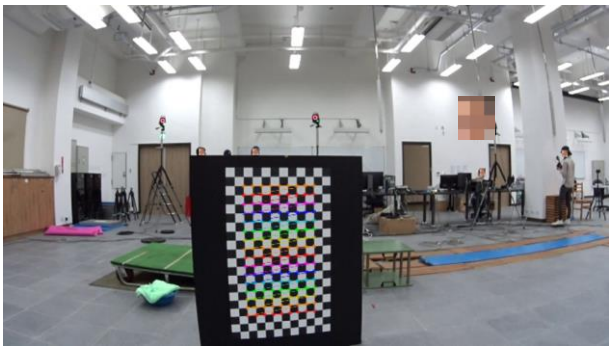


Figure 1: 12x20 checkerboard with a side length of 10 cm.

(a) (b)

Figure 2: (a) The distorted image. (b) The undistorted image.

2.4. 2D Ball Detector

Before detecting baseball, we need preprocess image shown in Figure 3, including: grayscale conversion and FDI (Frame Difference Image)[10]. Eq. (4) is the process of comparing the current frame with the previous frame to remove the background and morphology. Eq. (5) is the process to remove the image noise and Eq. (6) is to consider the brightness fo the ball.

$$FDI_n\left(x,y\right) = \begin{cases} 255, & \text{if } \left|I\left(x,y\right)_n - I\left(x,y\right)_{n-1}\right| > T \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

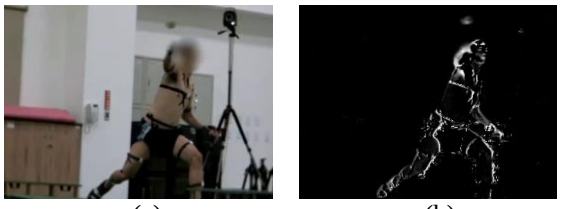
where *n* represents the frame index of the video sequence; *T* is the threshold of the pixel value difference; and $I\left(x,y\right)_n$ and $I\left(x,y\right)_{n-1}$ represent the current and previous frame pixel values, respectively.

$$PI_n = FDI_n \bullet SE = \left(FDI_n \oplus SE\right) - SE \quad (5)$$

PI (Preprocessed Image) is the inner product of *FDI* and *SE* (Structuring Element), which first dilates and then erodes [11].

$$BI_n\left(x,y\right) = \begin{cases} 255, & \text{if } PI_n > \text{gray threshold} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

BI_n (brightness image) Binding PI_n with gray threshold.



(a) (b)

(c) (d)

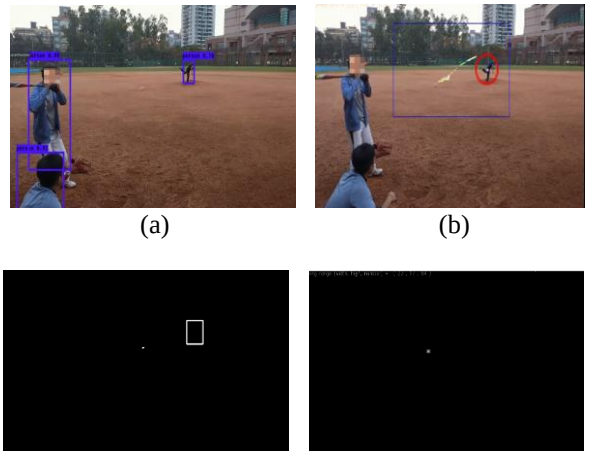
Figure 3: Preprocessed image. (a) Original image. (b) FDI. (c) Morphology. (d) Brightness.

We need to consider the feature of baseball including the radius and circularity ratio of the ball as expressed by Eq. (7), the velocity calculated by the Lucas-Kanade algorithm [12], and the specific area to detect the baseball in Figure 4.

$$\text{Circularity Ratio} = \frac{\text{Area in Circle}}{\pi r^2} \tag{7}$$

Figure 4: Ball detector: The blue area is the tracking area, and the color light path is the result of the optical flow algorithm.

However, in order to record the moment when the pitcher throws the baseball, the specific zone must include the ball and the pitcher. We use YOLOv3 [4]: Real-Time Object Detection to filter non-baseball objects in Figure 5.



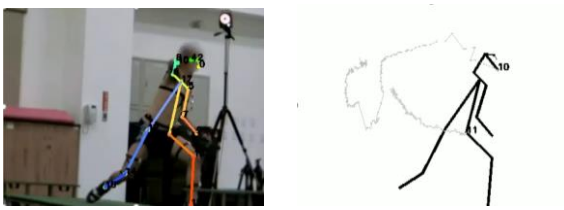
(a) (b)

(c) (d)

Figure 5: YOLOv3: Real-Time Object Detection: (a) YOLOv3 detector. (b) Use the ball detector and YOLOv3 detector. (c) The white rectangle is the YOLOv3 [4] that detects non-baseball objects in the blue tracking area. (d) The result of filtering the white rectangle.

2.5. 2D Person Pose Estimation

AlphaPose [2, 3] detects the person pose estimation as shown in Figure 6, including 0. nose, 1. left eye, 2. right eye, 3. left ear, 4. right ear; body: 5. left shoulder, 6. right shoulder 7. Left elbow, 8. Right elbow, 9. Right hand first, 10. Left hand first, 11. Left waist, 12. Right waist, 17. First, 13. Left knee, 14. Right knee, 15. Left foot, 16. Right foot. However, in order to select the right player, the longest body joint length is taken as the target, and finally multiple synchronized cameras are used to obtain joint points at different angles.



(a) (b)

Figure. 6: AlphaPose: (a) AlphaPose detector. (b) Filter the background and track specific joint point.

2.6 Three-Dimensional Space Reconstruction

To reconstruct a 3D structure from multiple views, we need to find the relative orientation between the cameras. The global 3D coordinate of the checkerboard in different images are calculated by the dichotomy to find the initial α and β and the depth h as shown in Figure. 7.

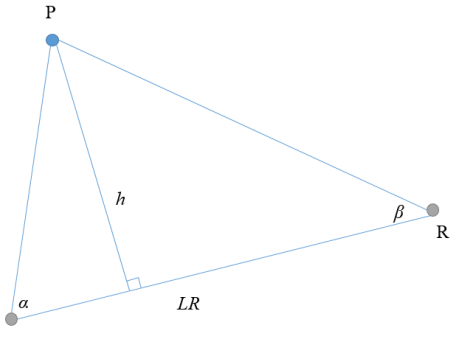


Figure 7: P is the calibration mark; L and R are the cameras; LR is the baseline of the two cameras; α, β is the angle between the camera and the calibration mark; and h is the vertical distance from the calibration mark to the baseline.

From the results of the calibration camera, we can calculate the camera's horizontal and vertical FOV (Field of View):

$$\begin{cases} FOV_h = 2\times\tan^{-1}\left(\dfrac{ch}{2d}\right) \\ FOV_v = 2\times\tan^{-1}\left(\dfrac{cw}{2d}\right) \end{cases} \tag{8}$$

FOV_h and FOV_v are horizontal and vertical FOV, respectively; d is the distance from the camera to the checkerboard, and ch and cw are the total width and height of the camera at distance d, respectively.

The FOV can calculate the parallax of all feature points in the image to obtain the corrected angles α and β as shown in Figure 8:

$$\begin{cases} \alpha = \alpha + d\theta_\alpha \\ \beta = \beta + d\theta_\beta \end{cases}, \quad d\theta = \tan^{-1}\left(\frac{2\cdot dp\cdot\tan\left(\dfrac{FOV}{2}\right)}{\dfrac{pl}{2}}\right) \tag{9}$$

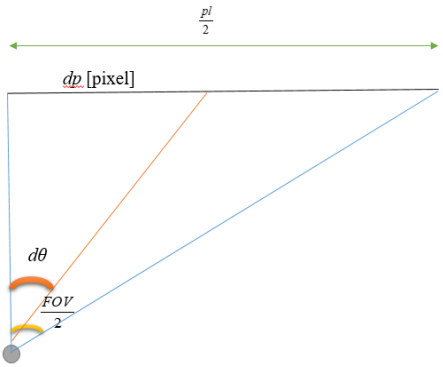


Figure 8: pl is the vertical or horizontal pixel length in the image; $d\theta$ is the angle of correction of different feature points; and dp is the distance (pixel) between the position of the feature point and the center line in the image.

By triangulation, we can get the y corresponding to each feature point:

$$y = \frac{LR\cdot\sin(\alpha+\beta)}{\sin\alpha\sin\beta} \tag{10}$$

Calculate the x, z of global 3D coordinate with the relationship of the triangle geometry and the result of the camera calibration.

$$x = \frac{2\cdot y\cdot p_x\cdot\tan\left(\dfrac{FOV_h}{2}\right)}{p_w} \tag{11}$$

$$z = \frac{2\cdot y\cdot\left(p_h - p_y\right)\cdot\tan\left(\dfrac{FOV_v}{2}\right)}{p_h} \tag{12}$$

p_x, p_y are the position of the joint point in the image; p_w, p_h are all pixels of length and width in the image.

In order to solve the problem of the shadow point on a single view, the multi-views feature point coordinates are projected onto the camera *C1* plane in Figure 9.

$$\begin{cases} x1 = x2\cdot\cos(\alpha_1+\beta_1) + y2\cdot\sin(\alpha_1+\beta_1) \\ y1 = x2\cdot\sin(\alpha_1+\beta_1) + y2\cdot\cos(\alpha_1+\beta_1) \end{cases} \tag{13}$$

$$\begin{cases} x1 = x3\cdot\cos(\alpha_3+\beta_3) + y3\cdot\sin(\alpha_3+\beta_3) \\ y1 = x3\cdot\sin(\alpha_3+\beta_3) + y3\cdot\cos(\alpha_3+\beta_3) \end{cases} \tag{14}$$

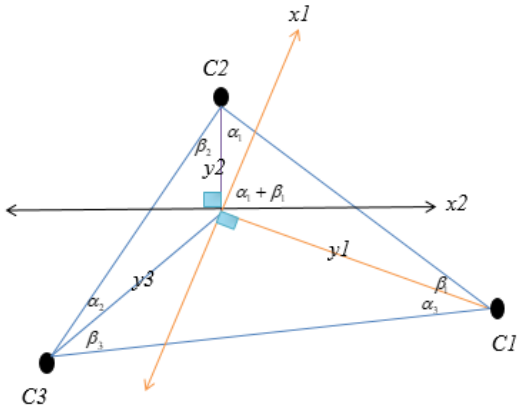
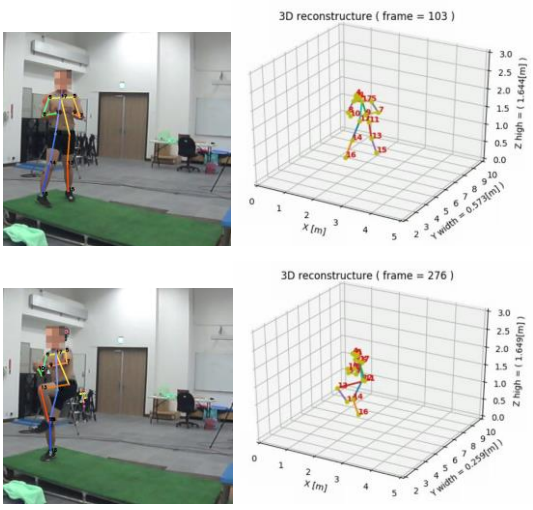
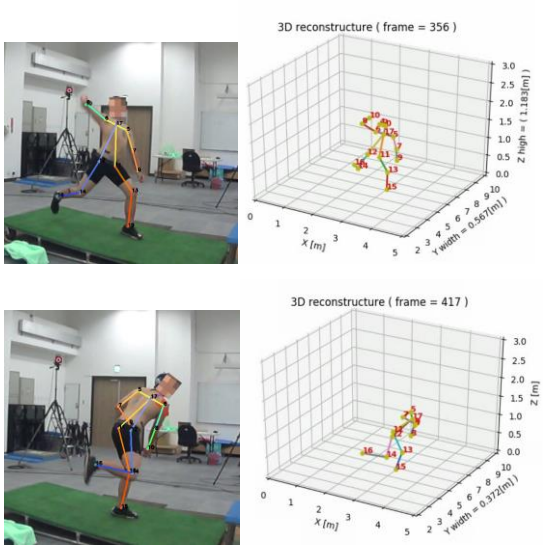


Figure 9: Multiple cameras and calibration marker in space, and C1, C2 and C3 are cameras.

3. EXPERIMENTAL RESULTS

Figure 10 shows the results of our 3D pose for baseball players at the National Taiwan University of Sports.

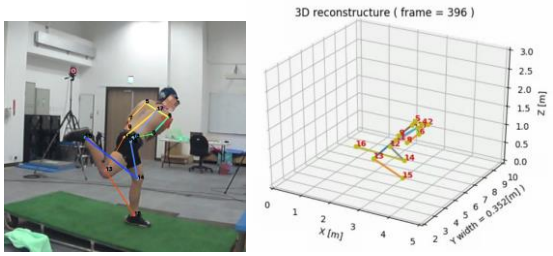




(a) (b)

Figure 10: (a) 2D pose estimation from AlphaPose. (b) 3D Pose Estimation.

The error of the model is measured by the 2D pose detector and the baseline between cameras. If the model gets wrong pose estimation from the 2D detector, it will get the wrong 3D pose estimation as shown in Figure 11.



(a) (b)

Figure 11: (a) A wrong 2D pose estimation from AlphaPose. (b) A wrong 3D pose estimation.

We use AlphaPose[2, 3] as a 2D pose detector because it is faster and more accurate than OpenPose[13-16] and Mask-RCNN[17] as shown in table 1. In our device, each camera can achieve 13 fps.

Table 1: The accurate and speed of AphaPose compare with OpenPose and Mask-RCNN methods.

Method	AP @Hard	fps
OpenPose (CMU-Pose)	32.3	5.3
Detectron (Mask R-CNN)	45.8	2.9
AlphaPose (PyTorch)	57.4	10.1

4. CONCLUSION AND DISCUSSION

A simple parallax theory is introduced into our model for 3D human pose estimation. The result is determined by the 2D joint point information of different cameras and the baseline measurement between cameras. As long as the camera parameters are fixed, it is practical in motion capture challenging scenes (e.g. outdoor sports). In addition, the method proposed in this study is not limited to baseball, but can be further extended to other sports to increase its variability and flexible.

5. FUTURE WORK

To increase the accuracy of the results, we will use VICON [1] to calculate the error and use the results of the multi-view to select the appropriate 2D pose detector results and retrain a 2D detector that belongs to the baseball.

ACKNOWLEDGEMENT

This work was funded by Ministry of Science and Technology (MOST) in Taiwan under Grants No. MOST 107-2627-H-028-002.

REFERENCES

[1] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A Study of Vicon System Positioning Performance," *Sensors*, vol. 17, p. 1591, 2017.

[2] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional Multi-person Pose Estimation," in *ICCV*, ed, 2017.

[3] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose Flow: Efficient Online Pose Tracking," in *BMVC*, ed, 2018.

[4] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv*, 2018.

[5] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ed, 1997, pp. 1106-1112.

[6] V. Chari and A. Veeraraghavan, "Lens Distortion, Radial Distortion," in *Computer Vision: A Reference Guide*, K. Ikeuchi Ed. Boston, MA: Springer US, 2014, pp. 443-445.

[7] D. C. Brown, "Close-range camera calibration," vol. 37, no. 8, pp. 855–866, 1971.

[8] J. G. Fryer and D. C. Brown, "Lens distortion for close-range photogrammetry," *Photogrammetric Engineering and Remote Sensing*, vol. 52, pp. 51-58, 1986.

[9] https://docs.opencv.org.

[10] S.-c. Cheung and C. Kamath, *Robust Techniques for Background Subtraction in Urban Traffic Video*. 2003.

[11] P. Soille, *Morphological Image Analysis: Principles and Applications*. Berlin, Heidelberg: Springer-Verlag, 2003.

[12] J.-Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm*. Intel Corporation Microprocessor Research Labs, 2000.

[13] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using Part Affinity Fields," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ed, 2017, pp. 7291-7299.

[14] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in *CVPR*, ed, 2017, pp. 7291-7299.

[15] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand Keypoint Detection in Single Images using Multiview Bootstrapping," in *CVPR*, ed, 2017, pp. 1145-1153.

[16] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ed, 2016, pp. 4724-4732.

[17] W. Abdulla, *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. Github, 2017.