A Simple Method to Calculate Positions in Pose Tracking to Verify Work Procedures

Manuscript received March 15, 2023; revised April 5, 2023

Kazumoto Tanaka
Faculty of Engineering
Kindai University
Higashi-hiroshima, Japan
kazumoto@hiro.kindai.ac.jp

***Abstract*— Although manufacturing processes are becoming increasingly automated, many factories still rely on manual operations. In such facilities, there is a strong need to automatically detect human errors by checking whether specified procedures are being followed. For this purpose, studies have been conducted on utilizing deep neural network (DNN) based three-dimensional (3D) human pose-tracking methods to examine work procedures. However, most of these techniques require a high-end computer equipped with a graphics processing unit (GPU). On the other hand, in this study, we adopt MediaPipe Pose, a lightweight pose estimation network provided by Google, to perform pose tracking on a low-end personal computer (PC) to enable such systems to be deployed in small factories. However, MediaPipe Pose cannot track the location of a human body because it estimates poses in a coordinate system with the waist as the origin (that is, in a root-relative coordinate system). Therefore, in this study, we developed a method to obtain the absolute coordinates of the root with a simple calculation. The results of an experimental evaluation show that the computational load of the proposed approach is negligible, and the repeatability of the estimation sufficed to evaluate a given operator's work on a predetermined working path. Therefore, the proposed methods enables work procedures to be checked using MediaPipe Pose on a low-end PC.**

Keywords*— *absolute root coordinates, deep neural network, pose tracking, root-relative coordinates, work procedure

I. INTRODUCTION

Automated logistics centers and smart factories have been commercialized mainly by large companies that can employ the latest robotic systems. However, many small factories still rely on manual operations, and there is a continuing need to train skilled workers. Therefore, many studies have been conducted to facilitate manual labor in industrial production or logistics settings, such as works on automatically detecting human errors by checking whether specified procedures are followed [1]-[4]. The key technology for verifying physical work procedures is that of action recognition methods comprising two key components, including human pose tracking and pattern matching between tracked and modeled data. In this study, we focus on pose tracking and propose a lightweight method for real-time tracking that can be implemented even on low-end personal computers (PCs).

For pose tracking, root-relative three-dimensional (3D) pose estimation methods have been developed to estimate the coordinates of key points on the human body (e.g., the joints) in a coordinate system the origin of which is set at a specific point of the human subject (e.g., in the center of their waist). The specific point and the local coordinate systems are referred to as root and root-relative coordinate systems, respectively. In addition, because many operations are performed while moving between multiple work positions (e.g., to operate a processing machine), pose tracking also requires estimating the position of the root, which changes as the subject moves. These methods for estimating the root-relative 3D pose and the absolute position of the root are becoming increasingly practical owing to advancements in deep neural network (DNN) models [5]-[10]. However, in most cases, a high-end computer equipped with a graphics processing unit (GPU) is required to execute such models, which are often too expensive to use in factories operated by smaller companies or with strict financial limitations. To overcome this problem, in this study, we utilize MediaPipe Pose [11] as a root-relative 3D pose estimation network that can be run nearly in real time on a low-end PC. However, MediaPipe Pose cannot estimate the absolute position of the root. Hence, in this study, we propose a simple calculation method to obtain the absolute position using the output results of MediaPipe Pose. This method enables pose-tracking methods to be implemented on low-end PC hardware. The contributions of this study are summarized as follows.

- We propose a simple computational method to obtain the absolute position of the root from root-relative coordinates. This method, together with a lightweight root-relative pose estimation network (e.g., MediaPipe Pose), is shown to be capable of estimating the absolute position of a pose nearly in real time even on a low-end PC.
- The accuracy of the position estimation was experimentally evaluated, and the results confirmed the practicality of using this method to verify adherence to specified work procedures.

II. RELATED WORK

First, we briefly review some key recent studies on manual operation recognition. The authors of [1] proposed a method to check whether a worker used the correct parts rack by estimating their 3D pose using a depth sensor (Microsoft Kinect) and tracking a colored glove using an RGB camera. Depth sensors generally employ infrared time-of-flight (ToF), and there is some concern that they may be affected by heat sources in factory environments. [2] and [3] utilized a common camera and YOLO [12], a DNN, for 2D pose estimation to evaluate work procedures. However, these methods used two-dimensional motion evaluations. The authors of [4] proposed

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/esl.v2i02.313

a method to evaluate drug-picking actions in 3D using Kinect and MediaPipe Pose. Because MediaPipe Pose estimates a human pose in a coordinate system with the waist as the origin, the absolute distance (depth) in the camera coordinate system cannot be obtained. Therefore, [4] a Kinect was used to estimate this distance. In contrast, the method proposed in the present work estimates depth by adding a simple calculation to the output of MediaPipe Pose.

The most common approach for estimating the absolute 3D-coordinates of keypoints for pose tracking is to combine the estimation of root-relative coordinates with the estimation of the absolute coordinates of the root. Studies on DNNs for estimating the root-relative coordinates of keypoints have largely achieved their aims, and the focus of recent research has shifted to estimating the absolute coordinates of the root. The authors of [5] proposed a network called RootNet that estimates parameters for estimating depth from the area of a bounding box indicating a person in an image. Similarly, the authors of [6] proposed a network called PoseLifter to estimate the canonicalized root depth with the focal length from an estimated 2D pose. More recent studies have further refined such estimation methods. For example, [7] proposed a deep regression network to estimate absolute 3D coordinates directly from images without intermediate 2D pose representations, and [8] improved the accuracy of poste estimations using a keypoint coordinate representation based on camera line-of-sight. Unlike these approaches, which estimate 3D poses from a single image, methods using multiple images of video frames have also been studied; for example, the authors of [9] used temporal convolutional networks (TCNs) to obtain absolute 3D human poses. While the above methods require a high-end computer with a GPU, the authors of [10] improved MobileNetV2 model [13] designed as a lightweight DNN to enable pose estimation on smartphones. However, [10] also combined a DNN trained to estimate root-relative coordinates with a model trained to estimate the absolute coordinates of the root. In contrast to all of the above methods, the proposed approach adopts a minimalist strategy to enable 3D pose estimation on low-end PCs. Our method replaces a DNN model for estimating the absolute coordinates of the root with a simple computation that does not require any network models.

III. Method

A. Calculation of Root 3D Coordinates

In this subsection, we describe the proposed method to calculate the 3D coordinates of a root represented by a camera coordinate system. Networks for root-relative pose estimation (MediaPipe Pose in this study) generally utilize a root-relative coordinate system with the same orientation as the camera coordinate system, and estimate both 2D (i.e., image) and root-relative 3D coordinates of the keypoints. Using these coordinate values, the perspective transformation equation that transforms the root-relative coordinates of a keypoint to image coordinates is given as follows in (1).

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix} (\boldsymbol{I} \quad \boldsymbol{t}) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \tag{1}$$

where $(u, v)^{\mathrm{T}}$, $(x, y, z)^{\mathrm{T}}$, f_x, f_y, $\boldsymbol{I}$, $\boldsymbol{t}$, s are the 2D coordinates of the keypoint (the origin is the principal point of the image), 3D coordinates, focal lengths along the x- and y-axes, an identity matrix, a position vector of the root expressed in the camera coordinate system, and a non-zero scalar, respectively (see Fig. 1). Assuming that the intrinsic parameters of the camera are known, $\boldsymbol{t}$ can be obtained by using the estimated keypoint coordinates as follows. Equation (2) is obtained by eliminating s in (1).

$$\begin{pmatrix} (u/f_x)z - x \\ (v/f_y)z - y \end{pmatrix} = \begin{pmatrix} 1 & 0 & -(u/f_x) \\ 0 & 1 & -(v/f_y) \end{pmatrix} \boldsymbol{t}. \tag{2}$$

The 2D vector on the left side and the 2 × 3 matrix on the right in (2) contain the 2D and 3D keypoint coordinates; therefore, if there are N keypoints, the vector and matrix are 2N-D and 2N × 3, respectively. Therefore, for 2 ≤ N, the matrix becomes a tall matrix and $\boldsymbol{t}$ is obtained by (3).

$$\boldsymbol{t} = (\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{v}, \tag{3}$$

where $\boldsymbol{v}$ and $\boldsymbol{M}$ denote the 2N-D vector and 2N × 3 matrix, respectively. The absolute pose coordinates (using the camera coordinate system) of each keypoint are obtained by adding $\boldsymbol{t}$ to the root-relative coordinates of each keypoint.

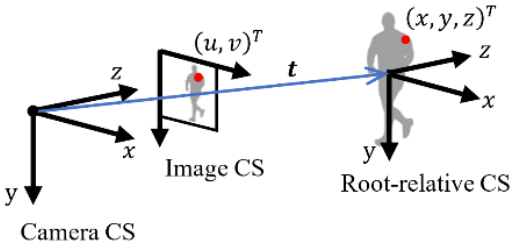

Fig. 1. Root-relative and related coordinate systems (CSs).

B. Keypoint Selection by RANSAC

In this subsection, we describe the proposed method to select keypoints from among visible keypoints for use in calculating $\boldsymbol{t}$. The results of some preliminary experiments showed that selecting keypoints close to the root improved the accuracy of $\boldsymbol{t}$. Hence, among the M keypoints close to the root, the proposed method selects a set of keypoints using the RANSAC algorithm such that the evaluation function of keypoint k defined in (4) is smaller than the given threshold e_{th}.

$$e_k = \left\| \begin{pmatrix} (u_k/f_x)z_k - x_k \\ (v_k/f_y)z_k - y_k \end{pmatrix} - \begin{pmatrix} 1 & 0 & -(u_k/f_x) \\ 0 & 1 & -(v_k/f_y) \end{pmatrix} \boldsymbol{t} \right\|, \tag{4}$$

where (u_k, v_k), (x_k, y_k, z_k), $\boldsymbol{t}$, $\|\cdot\|$ denote 2D coordinates of keypoint k, 3D coordinates of keypoint k, absolute coordinates of the root computed at the selected keypoints, and the L2-norm, respectively. The selection procedure with RANSAC is as follows.

Step 1) Create an empty list K for keypoints. Set a float-type variable e_{minsum} to the maximum value of the variable type and an integer-type variable c to 0.
Step 2) Calculate the L2-norm for all visual keypoints.
Step 3) Select M keypoints in order from that with the smallest L2-norm. /* M = 8 in the experiment.*/
Step 4) Randomly extract N keypoints from M keypoints. /* N = 4 in this experiment. */
Step 5) Calculate $\boldsymbol{t}$ by (3) with the N keypoints.
Step 6) For each of the N keypoints, compute e_k using $\boldsymbol{t}$ obtained in Step 5 and delete keypoints with e_k greater

than e_{th} (i.e., delete outliers). If there are no keypoints to delete, replace the keypoints in K with the N keypoints and go to Step 10. Otherwise, let N^* be the number of remaining keypoints (i.e., the number of inliers) and calculate $\boldsymbol{t}$ with the N^* keypoints /* e_{th} = 0.2 in the experiment. */

Step 7) For each of the N^* keypoints, compute e_k using $\boldsymbol{t}$ obtained in Step 6. If each e_k is less than e_{th} then replace the keypoints in K with N^* keypoints and proceed to Step 10; otherwise, let e_{sum} be the sum of each e_k.

Step 8) If e_{sum} is less than e_{minsum}, substitute e_{sum} for e_{minsum} and replace the keypoints in K with the N^* keypoints.

Step 9) Add 1 to c. If c is less than c_{max}, return to Step 4. /* c_{max} is the upper limit of the iterations, and c_{max} = 30 in the experiments. */

Step 10) Output K as the appropriate keypoint set. Then, terminate the procedure.

IV. Experiments and Results

A. Evaluation of Accuracy of Root Coordinate Estimation

The accuracy of the proposed estimation method was evaluated using the publicly available HumanEva-I [14] dataset for 3D human pose estimation. The intrinsic camera parameters are provided by the dataset and were used to compute (3) and (4). Assuming a situation in which a worker moves to multiple locations to work, the experiment evaluated the "walking" motions of the subjects in the dataset. An example of these results is shown in Fig. 2. The mean of the root position error (MRPE) was used to evaluate the estimated root coordinates. A comparison of the performance of proposed method and that of existing absolute 3D pose estimation models on HumanEva-I is presented in Table 1. In this study, Ray3D [8] was considered as a current state-of-the-art model.

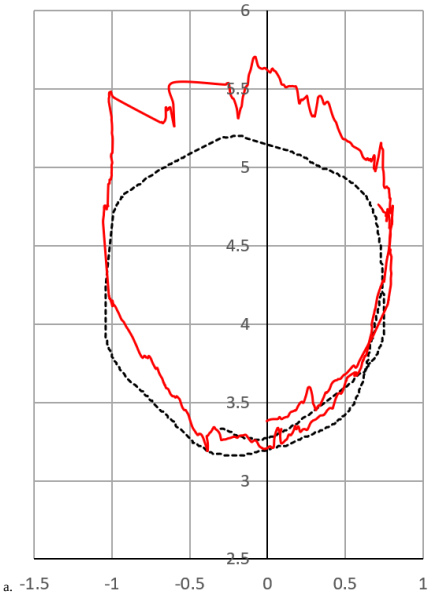

Fig. 2. Estimated root trajectory and ground truth for subject S1 with camera C1 in HumanEva-I. The red line represents the estimated result, and the black dotted line represents the ground truth.

TABLE I. MRPE COMPARISON

Method	***MRPE*** (mm)
PoseLifter [6]	1129.3*
Ray3D [8]	468.6
Proposed	497.2

* This is the result of a previous study [8], in which PoseLifter was retrained to compare root depth estimation.

B. Evaluation of Repeatability of Root Coordinate Estimation

Assuming the same situation as in the experiment described in Subsection A, the repeatability of the estimation of the proposed method was evaluated for a subject walking the same path repeatedly. The experiment was performed for the two paths shown in Fig. 3. The camera was placed such that the *xz* plane of the camera coordinate system (see Fig. 1) was horizontal. Each path was within 5.0 m of the camera, approximately 3.0 m long and about 2.5 m from each other. The estimated absolute coordinates (x, z) of the root after ten round trips for each of these paths are shown in Fig. 4. The first principal axis of the distribution in Fig. 4 was considered as the direction of the walking motion, and repeatability was evaluated by the standard deviation of the second principal axis. The standard deviation of the path closer to the camera (the path at left in Fig. 3) and the other path were 49.8 and 78.3 mm, respectively. The maximum widths in the second principal axis direction were 308.4 and 389.1 mm, respectively.

Finally, the framerate of the 3D pose tracking method was 15.5 frames per second (fps) on a PC running Windows OS, with an Intel Core i7-7660U CPU @ 2.50GHz and no GPU available, whereas root-relative 3D estimation without the absolute root estimation (i.e. without the proposed method) was also measured at 15.5 fps. Therefore, the proposed method added almost no load to the pose-tracking process.



Fig. 3. Two paths (double-headed arrow) for repeatability evaluation.

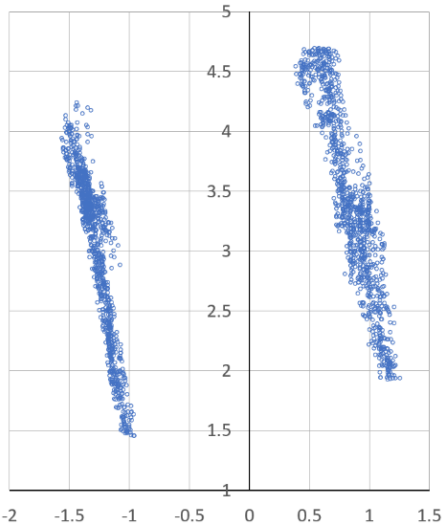

Fig. 4. Estimated x and z coordinates of root in two walks.

EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/esl.v2i02.313

V. Discussions

As shown in Fig. 2, the estimation accuracy was significantly lower at distances greater than about 5 m from the camera. This can be attributed to the conditions of the dataset conditions used to train the MediaPipe Pose model. As shown in Table 1, the MRPE was 497.2 mm, which was less than the accuracy of the state-of-the-art model. If this level of accuracy does not suffice in checking work procedures, the location of the camera must be adjusted, such as by keeping the distance between the operator and camera at less than 5 m.

Even with low accuracy of the root position estimation, a worker's path can be evaluated by comparison with previously examined working paths if the repeatability of the estimation is sufficiently high. Based on the standard deviation of the repeatability experiment, the magnitude of the error in the estimated coordinates was stationary, indicating high repeatability. The stationary magnitude of the error may be attributed to the experiment being conducted with the same person repeating the same action (walking). Therefore, these results show that the proposed method is effective for checking a given operator's work on a predetermined working path. In addition, Fig. 4 shows that the repeatability of the proposed method was better when the subject was closer to the camera.

No effect of the proposed calculation method on fps was observed. That is, fps is determined by the speed of the network model that estimates the root-relative coordinates (in this study, MediaPipe Pose). The computational complexity of the method is $O(n)$, where $n = c_{max} \times \mathrm{N}$ (c_{max} and N are the number of iterations and the number of keypoints, respectively, as described in Subsection B in III). Because the method can be used in combination with any root-relative coordinate-estimation network, the framerate should improve further if a lighter estimation network were used, for example, with the method provided in [10]. However, when performing pose estimation for multiple individuals, the calculation is performed for each root; thus, $n = c_{max} \times \mathrm{N} \times m$, where m is the number of persons. If the number of people to be tracked increases, the resulting effects on computational overhead may become an issue.

VI. Limitations and Conclusion

In this study, we have proposed a method for estimating the root coordinates of a person in a camera coordinate system from the output of a root-relative pose estimation network. The experimental results show that the system was able to evaluate a worker's path, and the computational load of the proposed method was negligible. Therefore, when used in conjunction with a lightweight root-relative pose estimation network (MediaPipe Pose in this work), work procedure-checking systems could be plausibly introduced into factory settings with minimal cost.

The calculations required by the proposed method involve the intrinsic parameters of the camera; however, existing simple methods can be used for this purpose. In the present work, the target of the work procedure-checking process was a single person. The development of methods to check multiple workers simultaneously remains as a topic for future research.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 21K12166.

References

[1] Y. Murata, Y. Takehara, Y. Uda, and T. Yoshikawa, “Picking and Assortment Operation Assistance Systems with the Depth Camera,” International Journal on Advances in Intelligent Systems, 11(3 & 4), pp. 299-308, 2018.

[2] C. Chena, T. Wanga, D. Lia, and J. Hongb, “Repetitive Assembly Action Recognition Based on Object Detection and Pose Estimation,” Journal of Manufacturing Systems, 55, pp. 325-333, 2020. https://doi.org/10.1016/j.jmsy.2020.04.018

[3] P. Lou, J. Li, Y-H. Zeng, B. Chen, and X. Zhang, “Real-time Monitoring for Manual Operations with Machine vision in Smart Manufacturing,” Journal of Manufacturing Systems, 65, pp. 709-719, 2022. https://doi.org/10.1016/j.jmsy.2022.10.015

[4] Y. Ono, O. D. A. Prima, “Assessment of Drug Picking Activity using RGB-D Camera,” Proc. ACHI2021, Nice, France, Jul. 2021.

[5] G. Moon, J. Y. Chang, and K. M. Lee, “Camera Distance-Aware Top-Down Approach for 3D Multi-person Pose Estimation from a Single RGB Image,” Proc. ICCV2019, Seoul, Korea, pp. 10133-10142, Oct. 2019.

[6] J. Y. Chang, G. Moon, and K. M. Lee, “Poselifter: Absolute 3D Human Pose Lifting Network from a Single Noisy 2D Human Pose,” arXiv:1910.12029v2, 2020. https://doi.org/10.48550/arXiv.1910.12029

[7] L. Jin, et al., “Single-Stage is Enough: Multi-Person Absolute 3D Pose Estimation,” CVPR2022, New Orleans, USA, 13076-13085, Jun. 2022.

[8] Y. Zhan, F. Li, R. Weng, and W. Choi, “Ray3D: Ray-Based 3D Human Pose Estimation for Monocular Absolute 3D Localization,” Proc. CVPR2022, New Orleans, USA, 13116-13125, Jun. 2022.

[9] Y. Cheng, B. wang, B. Y., and R. T. Tan, “Graph and Temporal Convolutional Networks for 3D Multi-person Pose Estimation in Monocular Videos,” Proc. AAAI-21, (virtual), 1157-1165, Feb. 2021.

[10] S. Choi, S. Choi and C. Kim, "MobileHumanPose: Toward Real-time 3D Human Pose Estimation in Mobile Devices," Proc. CVPRW2021, (virtual), pp. 2328-2338, Jun. 2021.

[11] MediaPipe Pose. https://github.com/google/mediapipe/blob/master/docs/solutions/pose.md (accessed on 1 March 2023).

[12] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A Review of Yolo Algorithm Developments,” Procedia Computer Science, 199, pp. 1066-1073, 2022.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proc. CVPR2018, Salt Lake City, USA, pp. 4510-4520, Jun. 2018.

[14] L. Sigal, A. O. Balan, and M. J. Black, “Humaneva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion,” International Journal of Computer Vision, 87, pp. 4-27, 2010.