A Multi-view Skeleton Data Fusion Method Based on BP Neural Network

1st Yueyi Li
School of Communications and Information Engineering
Chongqing University of Posts and Telecommunications
Chongqing 400065, China
s210131116@stu.cqupt.edu.cn

2nd Xin Su
Department of Electronic Engineering
Tsinghua University
Beijing, China
suxin@tsinghua.edu.cn

3rd Xibin Xu
Department of Electronic Engineering
Tsinghua University
Beijing, China
xuxb@tsinghua.edu.cn

***Abstract*—In recent years, human skeleton tracking technology has attracted a lot of attention in the fields of virtual reality, human-computer interaction and medical rehabilitation. Human skeleton tracking technology is the basis for building human models in virtual reality scenarios. Among them, Kinect camera is widely used as a motion tracking sensor for virtual reality human-computer interaction. However, many current studies on skeleton point tracking are limited to single or dual camera systems, which leads to problems such as occlusion, missing skeleton data and errors. To solve the problems of limited capture range and data occlusion of a single Kinect camera, this paper proposes a skeleton point tracking method based on multiple Kinect cameras. The method uses multiple Kinect cameras to track the 3D coordinates of 32 body joints simultaneously, and unifies the joint coordinates captured by multiple Kinect cameras in different viewpoints into the same world coordinate system through coordinate transformation. The BP (Back Propagation) neural network is used to train the skeleton data from multiple viewpoints, thus generating a reliable user skeleton position in real time. By this method, the problems of the existing methods for obtaining skeleton points in a single camera view are solved.**

***Index Terms*—Azure Kinect, Coordinate calibration, BP neural network, Data fusion**

I. Introduction

Virtual reality (VR) is a computer-based high-tech means for creating virtual environments that integrate the senses of sight, sound, and touch [1]. Virtual reality technology is widely used and plays an important role in cultural entertainment, educational training, engineering design, medical rehabilitation, and other fields. The creation of human body models is crucial to improve the interactivity and immersion of virtual reality human-computer interaction technologies. Using Kinect sensors to capture human motion is a method based on computer vision principles, which allows motion capture of the human body in the tracking area through a camera without the need to wear any equipment, with less constraint and a stronger sense of immersion.

Microsoft developed the Kinect sensor and its SDK to capture human motion and provide corresponding 3D joint coordinates in real time, without the use of any markers or handheld controllers. It is potentially cost-effective and can be widely used as a portable markerless motion capture tool for clinic and home-based gait observation. However, past studies have shown that the accuracy of motion tracking based on the first and second generation Kinect is low [2]. Microsoft Kinect [3] provides a low-cost and convenient method to acquire human posture by extracting the complete skeleton of human motion in real time for capture. The latest version of Azure Kinect DK has been released and is expected to improve image sensing technology. It features two selectable field-of-view modes, higher resolution, a more compact appearance, and weighs almost half of the previous version. However, using a single Kinect to recognize human pose often encounters problems such as missing data, errors, and occlusions [4]. As a result, we are unable to capture the human pose accurately. Although there are solutions to avoid these problems, acquiring and fusing data using multiple Kinects is a more reliable approach. However, how to obtain valid information through data fusion is still a challenge to be solved.

Many scholars have worked on information acquisition and data fusion accuracy of multiple Kinect sensors. Yunru [5] proposed a motion capture system based on dual Azure Kinect, which provides relatively accurate knee angles in terms of ground gait compared to standard 3D gait analysis systems, but still cannot solve problems such as data loss that cannot meet the need for high-precision recognition of human postures. Saputra [6] combined multiple human tracking modules in Azure Kinect based on the ONNX runtime, each using passive infrared as the human segmentation and recognition process and converting depth information to 3D positions, improving the accuracy of skeleton tracking, but the convenience needs to be improved. Liu [7] studied the data fusion technology for multi-sensor networks and proposed a data fusion model based on BP neural network to realize the system adaptive multi-source data fusion.

In this paper, we build a multi-view human skeleton tracking system to handle issues such as occlusion. We improve the performance of Azure Kinect skeleton tracking from several angles. By using a hardware arrangement scheme with three Kinects, we can transform the coordinate system of the skeleton data captured by the three Kinects to ensure the consistency and accuracy of the skeleton data. Then, by building a BP neural network using the Kinects' own

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

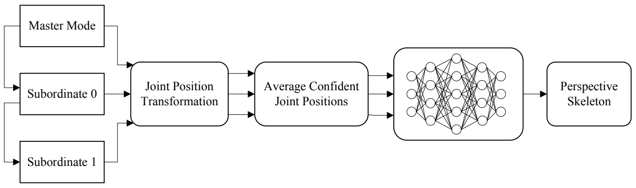

Master Mode
Subordinate 0
Subordinate 1
Joint Position Transformation
Average Confident Joint Positions
Perspective Skeleton

Fig. 1. Block diagram of the proposed system.

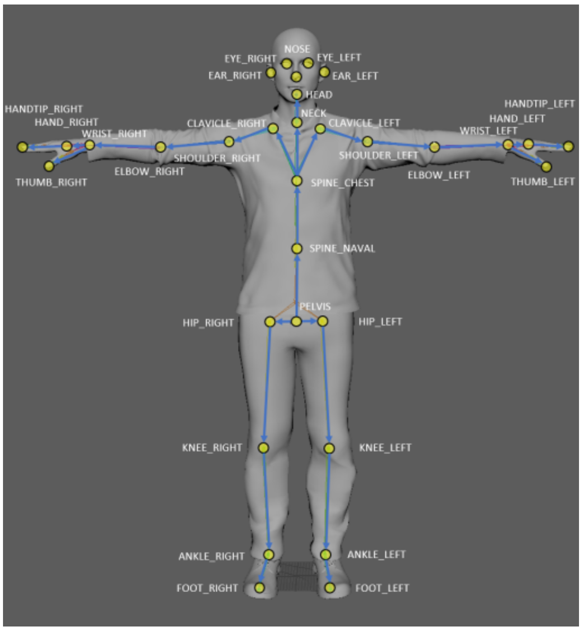


Fig. 2. Joints of skeleton that captured by Kinect sensor.

properties, we achieved the fusion of the three Kinect data by using the visual information captured by the Kinects as the input and output samples of the neural network. The training of the neural network provides a reliable skeleton position, solving the problems of missing data, errors and occlusions when skeleton data is collected by a single Kinect.

II. System Design

This section describes a skeleton data fusion system based on BP neural network, as shown in Fig. 1. The system consists of three Kinect sensors, which are each connected to a personal computer through a universal serial bus (USB). The user's skeleton information are collected by each computer and transmitted to the main server for data processing via Transmission Control Protocol /Internet Protocol (TCP/IP). If motion sensors are detected, skeleton information from each sensor is transmitted to the main server for processing. To integrate the skeleton data with constant body size, orientation and root position of the user, we used coordinate calibration and skeleton normalization methods to overcome these problems. Then, the predicted skeleton information can be derived based on the skeleton data collected by the three cameras as input to the neural network. Each process is described in detail in the following subsections.

A. Coordinate System Transformation

Azure Kinect is the latest generation of Kinect products from Microsoft, which is specifically designed for developers. The device includes a depth sensor, RGB camera, 7-microphone array, accelerometer and gyroscope, and pins for external synchronization [8]. Each Kinect sensor can obtain joint information of 32 human bones at 30 frames per second(fps) through the Kinect Software Development Kit (SDK) provided by Microsoft. The human skeleton information consists of the tracking state and position values of the bones. These skeleton joints are composed in a certain hierarchical structure, distributed from the center of the body toward the extremities. Each skeleton joint (articulation) connects a parent joint to a child joint. Fig. 2 shows the joint locations and the connections relative to the body.

When performing data acquisition with multiple cameras, we need to synchronize the cameras with each other in order to ensure that each frame of data captured by each camera is from the same moment of the scene. When using multiple Kinect devices for connection, we divide them into two attributes: master and subordinate. There can be only one master device, while the rest of the devices are subordinate. To achieve synchronous operation, we connect the input port of the subordinate devices to the output sync port of the master device via a splitter. In this way, the master device sends a synchronization signal to ensure that all cameras start collecting data at the same moment. In the actual scenario used in this paper, we used three cameras for data acquisition and arranged them in the arrangement shown in Fig. 3. This ensures that the data captured by each camera is at the same point in time and provides consistency for subsequent data processing.

When three Kinect devices are used to capture human postures, each Kinect is able to acquire skeleton joint point data of the target from a different viewpoint, and each Kinect has its own coordinate system. Therefore, these cameras need to be calibrated to translate their captured human pose into the same spatial coordinate system. The purpose of camera calibration is to obtain external parameters between cameras without internal calibration, because the motion capture module in this paper only involves skeleton coordinates and does not involve conversion between depth and color images. To achieve camera calibration, a checkerboard grid coordinate calibration method can be used by acquiring the rotation and translation matrices between multiple cameras [9]. The skeleton joint point data acquired by each Kinect is transformed by coordinates to a unified world coordinate system and then transferred to a data fusion server for subsequent processing. The ultimate goal is to establish a nonlinear mapping relationship of the skeleton points collected by three cameras from the 2D image coordinates.

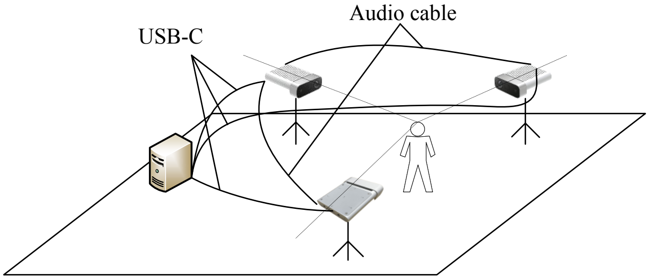


Fig. 3. Schematic illustration of the Three Kinect System.

Using the equation for the intrinsic matrix of the camera:

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x&0&c_x\\0&f_y&c_y\\0&0&1\end{bmatrix}\begin{bmatrix}r_{11}&r_{12}&r_{13}&t_1\\r_{21}&r_{22}&r_{23}&t_2\\r_{31}&r_{32}&r_{33}&t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \quad (1)$$

In the pixel coordinate system, $[u; v; 1]$ represents the coordinates of the projection point, while in the world coordinate system, $[X; Y; Z; 1]$ represents the position coordinates of the object. Transformation of the object from the world coordinate system to the camera coordinate system can be achieved by rigid body transformation, a process in which no deformation of the object occurs and only rotation and translation are involved. The rotation matrix $\mathbf{R}_{3\times3}$ and the translation vector $\mathbf{T}_{3\times1}$ are external parameters describing the world coordinate system to the camera coordinate system, also known as the external reference matrix of the camera, and they describe the pose of the camera in the world coordinate system. Assuming that the current scaling factor is s=1, if the image from the camera is scaled, then the other parameters should be scaled by the same ratio.

To obtain the desired features of the target image and use it as a tool for calibrating the coordinate systems between the master and slave devices, we can print a black-and-white chessboard pattern. One side of the chessboard pattern should consist of an equal number of black squares, while the other side should have an equal number of white squares [10]. By utilizing such an image, we can perform calibration between the master and subordinate devices, establishing a consistent coordinate system relationship between them.

B. Skeleton Data Normalization

Since the skeletal data collected by the Kinect sensors belong to different coordinate systems, in order to integrate these skeletal data, calibration is first required to obtain the camera matrix of each Kinect sensor. Then, for the coordinates of each joint position, it needs to be normalized so that it has a consistent scale in the same coordinate system.

The body size of the user can have an impact on the position of the skeletal joints, as different body sizes can lead to differences in joint positions. Therefore, normalization is required when fusing skeletal joint position data to ensure that body sizes remain consistent across different human bodies [11]. Therefore, an ergonomic kinematic tree consisting of 32 joints is considered to describe the structure of skeletal joints. In the kinematic tree, joints are represented as nodes and limbs are represented as edges. According to the definition of the kinematic tree, we designate the Spine Base joint ($j = 0$) as the root node of the tree, while the other 31 joints are used as branch nodes.

Let L_l represent the length of the l_{th} limb, $l \in 1, 2, 3, ..., 32$, m_l and n_l represent the starting and ending joints. For $j = 0, 1, 2, ..., 32$, the fused skeletal position is $\mathbf{P}_j$, and let $\mathbf{N}_j$ denote the coordinates of the j_{th} joint of the normalized skeleton. We can determine $\mathbf{N}_j$ by a normalization process. The normalization process starts with the root joint (Spine Base joint). The root normalizes the joint coordinates of the fused skeleton as:

$$\mathbf{N}_{n_l} = \mathbf{N}_{m_l} + L_l \cdot \frac{\mathbf{P}_{n_l} - \mathbf{P}_{m_l}}{\|\mathbf{P}_{n_l} - \mathbf{P}_{m_l}\|} \quad (2)$$

The user's position in the world coordinate system will change according to their own skeletal movements and the tracking results will be inaccurate. To ensure that the absolute position of the skeletal data in the world coordinate system remains constant for the user, we need to choose a fixed coordinate origin as the reference point [12]. Spine Base joint is used as the origin of the world coordinate system, and the positions of the other joints are transformed accordingly as:

$$\mathbf{N}_j = \mathbf{N}_j - \mathbf{N}_0, j = 1, 2, 3, \ldots, 32 \quad (3)$$

C. Skeletal Data Fusion based on BP Neural Network

The BP neural network is a multi-layer feed-forward neural network, which simulates the working principle of human brain and trains the network with a certain amount of data samples to convert the complex logical relationship between the data into the connection weight parameters between the network nodes.The BP neural network consists of nonlinear transformation units, which use the mechanism of error back propagation to realize the nonlinear mapping from The BP neural network consists of nonlinear transformation units that use the error back propagation mechanism to achieve a nonlinear mapping from input signals to output patterns. The basic algorithm is the least squares method, which calculates the weight value that minimizes the error between the network output and the desired output by the gradient search method.The learning process of BP neural networks consists of two main stages: the first stage is the forward propagation of the network, where the input information of the network is processed by the hidden layer; the second stage is the backward propagation of the network [13]. In the back propagation stage, when the output of the output layer does not match the desired output, we calculate the difference (error) between the desired output and the actual output by recursively calculating layer by layer, and then dynamically adjust the weight parameters according to this difference. By continuously iterating this process, we can obtain the ideal fusion data so that the output is gradually close to the desired output.

The training process of the neural network is as follows: the number of nodes in the input layer n, the number of nodes in the hidden layer l and the number of nodes in the output layer m are determined according to the input-output sequence $[X, Y]$, the connection weights from the input layer to the hidden layer is w_{ij}, the connection weights from the hidden layer to the output layer is w_{jk}, the bias of the hidden layer is a_j, the bias of the output layer is b_k, the learning rate is η, and the activation function uses the Sigmoid function:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

The output of the hidden layer neurons as:

$$H_j = g\left(\sum_{i=1}^{n} w_{ij} x_i + a_j\right) \quad (5)$$

The output of the output layer neurons as:

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

$$O_k = \sum_{j=1}^{l} H_j w_{jk} + b_k \tag{6}$$

Y_k denotes the desired output value of the neural network, calculate the error between them as:

$$E_k = Y_k - O_k \tag{7}$$

Then, updated the connection weights and the bias, as follows:

$$w_{ij} = w_{ij} + \eta H_j (1 - H_j) x_i \sum_{k=1}^{m} w_{jk} E_k \tag{8}$$

$$w_{jk} = w_{jk} + \eta H_j E_k \tag{9}$$

For each sample, learning is performed by adjusting the connection weights and biases until all samples have completed learning. During each iteration, it is determined whether the global output error function reaches a set convergence limit. If the error function reaches the predefined error, the iteration is stopped. In this way, it is ensured that the error is gradually reduced during the learning process until the desired convergence state is reached.

The loss function uses mean square error MSE:

$$MSE = \frac{\sum_{i=1}^{n} \left[(x_i - x_i^{'})^2 + (y_i - y_i^{'})^2 + (z_i - z_i^{'})^2 \right]}{n} \tag{10}$$

where x_i, y_i, z_i are the true values of the i^{th} data in one training, $x_i^{'}$, $y_i^{'}$, $z_i^{'}$ are the predicted values given by the neural network.

The error is back-propagated using a back-propagation algorithm to update the neural network parameters. During the training, we divide the data set into a training set and a validation set, and use the validation set for model tuning in order to avoid overfitting.

III. Experimental Results and Analysis

A. Parameter Setting

After several experiments, the parameter settings for running the program are shown in TABLE I:

TABLE I
Command Line Parameter

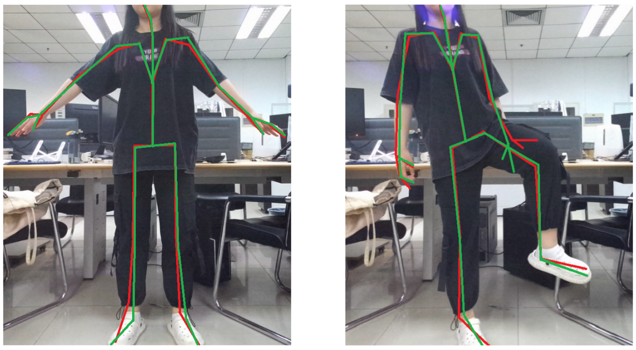
Parameter	Value
chessboard height	9
chessboard width	6
chessboard square length	15
depth threshold (mm)	1000 (default)
color exposure (ms)	8000 (default)
powerline frequency (Hz)	60 (default)
Number of hidden layers	3

B. Experiment for Convinced Skeleton

In the three-view skeleton fusion system, information related to Kinect sensors was provided by the proposed method at 30 fps. The three-view skeleton fused in real time after calibration synchronously with Kinect sensors.

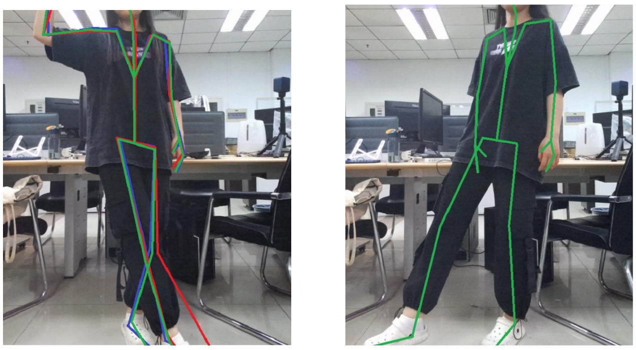
If only a low confidence level exists for the joint, the available data from other devices can be used. The pose captured by a single camera may produce incorrect skeleton positions due to occlusion, and with the addition of the devices, the angles of the other devices can compensate for the deficiencies of the single camera. As can be seen in Fig. 4 (a), after coordinate calibration, the skeleton output by the two devices in the unobstructed case have largely overlapped, and the hand have occlusions in Fig. 4 (b), thus resulting in an inaccurate skeleton position output by one of the devices.

Fig. 5 (a) shows a system with three devices, where one leg is occluded due to crossover, resulting in an inaccurate bone position in one view. But overall, the leg occlusion in one view has less impact in the overall system, and using three devices will have more confident joint positions. Training with a BP neural network requires the average of the skeleton data collected by the three devices to be used as training data. In the neural network, the weights of the occluded joint points are reduced to obtain more accurate skeleton data. Fig. 5 (b) shows the skeleton prediction after the training of the neural network, and it can be seen that the occluded hand is still able to have good skeleton tracking position. Further, we conducted more experiments and found that the three devices can solve the occlusion problem of any part of the body.



(a) No occlusion (b) Hands occlusion

Fig. 4. Skeleton recognized by double Kinects.



(a) Three devices (b) Neural network prediction

Fig. 5. Skeleton recognized by three Kinects and prediction.

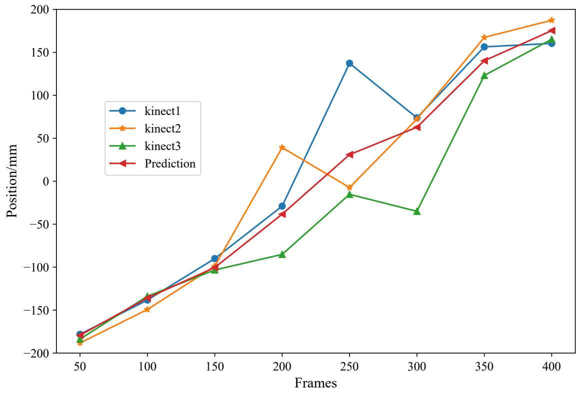


Fig. 6. Coordinate Fusion of Hand Joints.

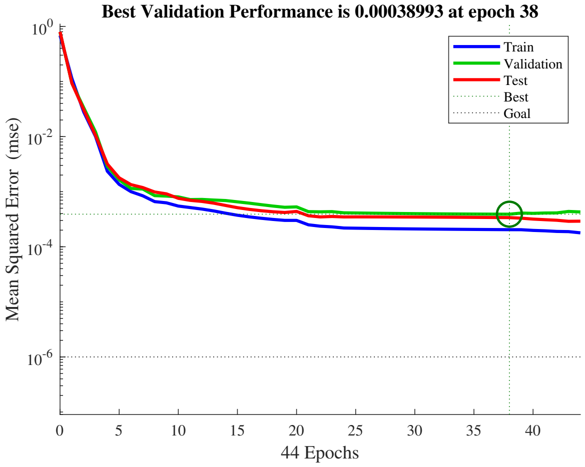


Fig. 7. Loss of Training and Validation.

This experiment verifies the effectiveness of the data fusion algorithm based on BP neural network in compensating the problem of poor recognition accuracy due to occlusion. The hand coordinates were selected for the experiment, and after the data were collected by the three Kinect respectively, the corresponding predicted data were obtained by the data fusion algorithm from the data after the coordinates were unified, as shown in Fig. 6.

The data fusion of the three Kinect data is effectively fused in the frames with missing and occluded data, and the prediction generates a set of more continuous posture coordinate data. The effectiveness of the data fusion algorithm in this paper is demonstrated. At the same time, the credibility of the acquired data is improved by data fusion, which makes the human posture recognition more accurate.

C. *Evaluation*

During the experiment, we set the learning rate to 0.01. According to Fig. 7, the root mean square error of the above neural network training reached a satisfactory convergence accuracy of about 0.00038993 after about 40 iterations. From this experimental result, it is clear that the multi-Kinect camera system solves the problem of insufficient view angle of a single camera; meanwhile, the accuracy of data fusion using BP neural network method is sufficient to meet the requirements of virtual reality vision system.

Conclusion

In this paper, we propose a skeleton data fusion method based on BP neural network, which aims to reduce the occlusion problem during human skeleton tracking by using Kinect devices to collect data from multiple viewpoints. To achieve coordinate calibration of multiple cameras, we use the tessellation method. Then, the skeleton data collected from different devices are integrated by a BP neural network training method to further improve the accuracy and real-time of skeleton point tracking. The experimental results show that the method can effectively render the skeleton motion of the characters and provide reliable skeletal data for driving the character models in virtual reality scenes. In the future, we will further improve the method and apply it to more types of virtual scenes or somatic rehabilitation training games to enhance user experience and rehabilitation effects.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2020YFB1806702).

References

[1] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, “Toward characteristic-preserving image-based virtual try-on network,” in *European Conference on Computer Vision (ECCV)*, 2018, pp. 589–604.

[2] R. A. Clark, B. F. Mentiplay, E. Hough, and Y. H. Pua, “Three-dimensional cameras and skeleton pose tracking for physical function assessment: A review of uses, validity, current developments and kinect alternatives,” *Gait posture*, vol. 68, pp. 193–200, 2019.

[3] M. Tölgyessy, M. Dekan, L. Chovanec, and P. Hubinskỳ, “Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2,” *Sensors*, vol. 21, no. 2, p. 413, 2021.

[4] N. Li and X. Zhao, “A benchmark for gait recognition under occlusion collected by multi-kinect sdas,” *arXiv preprint*, 2021.

[5] Y. Ma, B. Sheng, R. Hart, and Y. Zhang, “The validity of a dual azure kinect-based motion capture system for gait analysis: A preliminary study,” in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 1201–1206.

[6] A. A. Saputra, A. R. A. Besari, and N. Kubota, “Human joint skeleton tracking using multiple kinect azure,” in *2022 International Electronics Symposium (IES)*. IEEE, 2022, pp. 430–435.

[7] L. Shuai, “Multi-sensor data fusion algorithm based on bp neural network,” in *Journal of Physics: Conference Series*, vol. 1584, no. 1. IOP Publishing, 2020, p. 012025.

[8] J. A. Albert, V. Owolabi, A. Gebel, C. M. Brahms, U. Granacher, and B. Arnrich, “Evaluation of the pose tracking performance of the azure kinect and kinect v2 for gait analysis in comparison with a gold standard: A pilot study,” *Sensors*, vol. 20, no. 18, p. 5104, 2020.

[9] T. H. M. Siddique, Y. Rehman, T. Rafiq, M. Z. Nisar, M. S. Ibrahim, and M. Usman, "3d object localization using 2d estimates for computer vision applications," in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. IEEE, 2021, pp. 1–6.

[10] N. Eichler, H. Hel-Or, and I. Shimshoni, "Spatio-temporal calibration of multiple kinect cameras using 3d human pose," *Sensors*, vol. 22, no. 22, p. 8900, 2022.

[11] B. Kwon, J. Kim, K. Lee, Y. K. Lee, S. Park, and S. Lee, "Implementation of a virtual training simulator based on 360° multi-view human action recognition," *IEEE Access*, vol. 5, pp. 12 496–12 511, 2017.

[12] B. Yao, H. Gao, and X. Su, "Human motion recognition by three-view kinect sensors in virtual basketball training," in *2020 IEEE REGION 10 CONFERENCE (TENCON)*. IEEE, 2020, pp. 1260–1265.

[13] F. Utaminingrum, S. J. A. Sarosa, C. Karim, F. Gapsari, and R. C. Wihandika, "The combination of gray level co-occurrence matrix and back propagation neural network for classifying stairs descent and floor," *ICT express*, vol. 8, no. 1, pp. 151–160, 2022.