Computer Science and Information Systems 22(4):1687–1705 https://doi.org/10.2298/CSIS241002067L

A Study of Real-Time Operations by Converting Human Skeleton Coordinates to Digital Avatars

Fei-lung Lin[1], Jui-Hung Kao[2], Yu-Yu Yen[3,4,⋆], Kuan-Wen Liao[5], and Pu Huang[6,⋆]

[1] Institute of Technical and Vocational Education, National Taipei University of Technology, Taipei, Taiwan
t110499005@ntut.edu.tw
[2] Department of Information Management, Shih Hsin University, Taipei, Taiwan
kjhtw@mail.shu.edu.tw
[3] Center of General Education, Shih Hsin University, Taipei, Taiwan
melyen@mail.shu.edu.tw
[4] Department of Biomedical Engineering, National Yang Ming Chiao Tung University, Taipei, Taiwan
sheepkelly19.be11@nycu.edu.tw
[5] Department of Information Management, Shih Hsin University, Taipei, Taiwan
m111660002@mail.shu.edu.tw
[6] School of political science and law, Shaoguan University, Shaoguan, China
20201047@sgu.edu.cn

Abstract. This study aims to develop a real-time motion recognition system that translates skeletal human movements into a virtual environment. This will be achieved through the use of advanced tech-niques for the accurate capture of human skeletons and coordinate conversion. This paper investi-gates the acquisition and processing of motion data for virtual characters using depth cameras to obtain depth information. This study identifies six specific actions: left kick, right kick, left punch, right punch, squatting, and sitting. The experimental process successfully integrated RGB+D cameras, Media Pipe, and OpenCV into Unreal Engine models to capture and display human skeletal and joint positions in real-time. The experimental results show that the system achieved a precision of 100% for all motion detections, with an accuracy of more than 94%. How-ever, the recall rate for specific actions was lower, reaching 88%.

Keywords: Mixed Reality, Confusion Matrix, Motion Recognition.

1. Introduction

The application of image recognition technology has rapidly become a critical aspect of modern tech-nology, with the capacity to simulate and even surpass the capabilities of the human visual system. There is potential for advancement in several fields, including virtual reality (VR), augmented reality (AR), autonomous driving, and intel gent surveillance. Technology has the potential to enhance efficiency across a range of industries and

⋆ Corresponging authors

expand the scope of applications, thereby contributing to the advancment of these sectors. Nevertheless, accurately capturing and converting between physical and virtual environments remain a significant challenge, particularly in integrating digital avatars. While image recognition technology has demonstrated considerable success across various domains, technical challenges still exist to achieve higher precision and a more comprehensive application.

The increasing prevalence of mobile devices, such as smartphones and tablets, has led to a surge in demand for rapidly identifying captured subjects. This can be achieved by using cameras to scan (QR) codes, which can be used to obtain URLs or product information. Alternatively, document capture can facilitate text recognition and scanning functions. The aforementioned functionalities are made possible by image recognition technology, which also plays a pivotal role in the advancement of fields such as virtual reality (VR) and augmented reality (AR). VR technology enables the simulation of an entirely virtual environment, facilitating immersive user interactions. In contrast, AR technology overlaps virtual information with the real world, thereby achieving a fusion of reality and virtual elements.

The fundamental aspect of both technologies is the perception and recognition of the surrounding environment. This is paramount for determining the position and orientation of virtual objects, enabling interaction with the real world. This study aims to develop novel methodologies and techniques for precisely and accurately capturing the human skeleton. This process involves a thorough investigation of different types of sensors and imaging devices, instilling confidence in the research process. The selection and optimization of hardware is also a key part of this study, ensuring stable operation in a variety of scenarios. The proposed enhanced algorithms will improve the precision and practicality of the capture process. Furthermore, the study examines methods to incorporate additional human characteristic values to improve the reliability of skeletal capture. Another significant challenge is the efficient conversion of coordinate information into virtual space. Virtual space modeling techniques facilitate the real-time translation of physical human skeletal coordinates into actions within a virtual environment. This includes the calibration and transformation of coordinate systems to achieve consistency and accuracy. In addition, the data processing workflow was optimized to facilitate the efficient transformation of coordinates, thereby enabling real-time synchronization.

The system developed through this research is expected to open new avenues for virtual reality and gaming applications and extend to various fields such as social networks, remote conferencing, and virtual performances. The system enables users to interact in real-time within virtual spaces, participating in diverse activities through digital avatars. Technology has the potential to offer not only a novel entertainment experience but also commercial value and social impact. This could be achieved through technology to communicate with friends, attend virtual meetings, or perform online.

2. Materials and Methods

2.1. Human Motion Recognition

The field of computer vision has long been concerned with the recognition of human motion. Traditional methods have employed contour detection techniques to track the

human body and infer movements by calculating the torso's range of motion [1]. However, these conventional approaches have certain limitations, particularly their susceptibility to variations in camera angles and environmental backgrounds, which can lead to reduced stability in practical applications.

Several neural network models have been developed to recognize human motion. Among these models, the 2D Convolutional Neural Network (2DCNN)[2], which encompasses Convolutional Neural Networks (CNN) and Two-Stream Networks, has been extensively utilized in image recognition. However, these models are limited in processing image sequences with temporal information, which prevents them from in erring the temporal order of actions. To address this issue, some researchers have proposed recurrent neural network models capable of simultaneously learning temporal and spatial features. These include recurrent neural networks (RNN) [3], long-short term memory networks (LSTM) [4], and gated recurrent units (GRU) [5]. Furthermore, there are 3D Convolutional Neural Network (3DCNN) models [6] , including Inflated 3D ConvNet (I3D) [7] , Pseudo-3D Convolutional Neural Network (P3D) [8], and Separable 3D Convolutional Neural Network (S3D) [9], as well as models that incorporate attention mechanisms. However, these models typically have many parameters and high computational costs, which presents a significant challenge for their practical application in real-world scenarios.To achieve an optimal balance between accuracy and computational cost, researchers have proposed the implementation of new model architectures. The trade-off above models includes the first convolutional network (FstCN)[10], the temporal relation network (TRN) [11], the efficient channelized video model (ECO) [12], the multi-mode fusion network (MFNet), and the R (2 + 1) D convolutional neural network (R (2 + 1) D). These models have been designed to achieve high accuracy while maintaining computational efficiency, making them more suitable for practical human motion recognition applications. One motion recognition method, the Temporal Shift Module (TSM), is depicted in the accompanying illustration. This model can achieve the performance of 3DCNN models while only requiring the computational complexity of 2DCNN model.

The fundamental concept is to extract spatial features through a 2DCNN architecture, subsequently shifting the feature maps of selected channels by one frame in either the forward or backward direction of the temporal dimension. This approach integrates temporal and spatial information without needing temporal convolution operations. This method enables simultaneous learning of temporal features while preserving the integrity of spatial features, and it does so with a relatively low computational cost.

A motion recognition system solely focused on spatial and short-term temporal features may be susceptible to overlooking the significance of long-term temporal information for accuracy. Consequently, a neural network framework called Temporal Segment Network (TSN) [13] was devised to capture long-term motion information. The input video sequence is divided into K segments, with a randomly selected snippet extracted from each. Subsequently, each snippet is processed through a convolutional neural network, generating K classification scores. The above scores are subsequently integrated to create the final recognizable human action.

In video data processing, most CNN models are designed with the primary objective of two-dimensional image analysis. A straightforward approach treats the video as a sequence of static images, applying a 2DCNN for frame-by-frame recognition. However, this approach cannot capture motion information in the temporal domain. To effectively

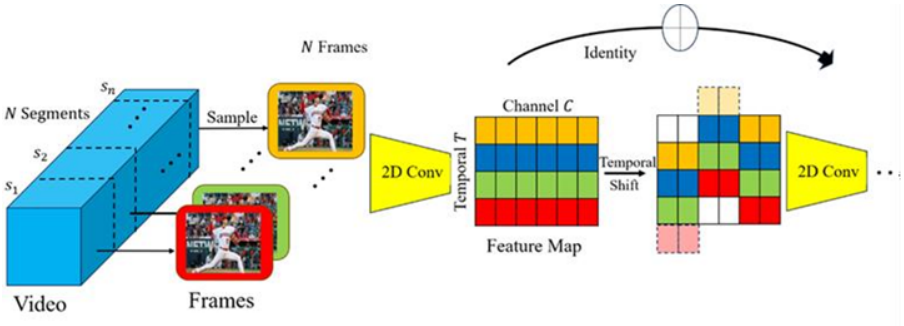

Fig. 1. TSM Architecture Diagram

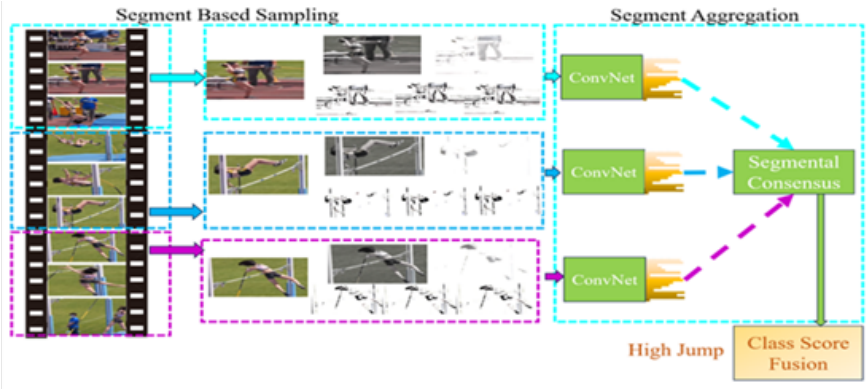

Fig. 2. TSN Architecture Diagram

incorporate temporal details, this study explores the use of 3DCNN to recognize human actions. The operation of a 3DCNN on the input video simultaneously learns spatiotemporal features, providing a more accurate representation of the video.

In the context of 3DCNN, small segments are extracted from consecutive input video frames for processing. This is exemplified in the following diagram. The convolution operation is performed by sliding a three-dimensional kernel over the input, simultaneously modeling temporal and spatial features. The data used by 3DCNN is sequential, typically comprising multiple frames of a video or a series of integrated segmented images. The input data is represented as a 3D cube, and the convolutional kernel is also a cube. The convolutional kernel performs sliding-window operations over the input data's spatial dimensions (length, width, and depth) to compute inner products, resulting in a single value in the output data. In the convolutional layer, each feature map is connected to multiple adjacent consecutive frames from the previous layer. This enables the capture of motion information. The formula for a 3D CNN is as follows:

$$\nu_{ijk}^{L} = \tanh\,[2061?](b^{1} + \sum_{m} \sum_{p=0}^{p-1} \sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} \varpi_{pqr}^{m} \bullet v_{(i+p)(j+q)(k+r)}^{l-1}) \tag{1}$$

In this formula, ν_{ijk}^{L} represents the feature map value at position ([2148?],j,k) in the I-th layer, bI denote the bias term of the I-th layer, [D835?][DF14?][D835?][DC5A?] signifies the 3D convolution kernel weight of the m-th feature map in the (I−1)-th layer, and $\nu_{(i+p)(j+q)(k+r)}^{\iota-1}$ represents the feature map value at position ([2148?]+p , j+q , k+r) in the (I−1)-th layer. The tanh function is used as the activation function.

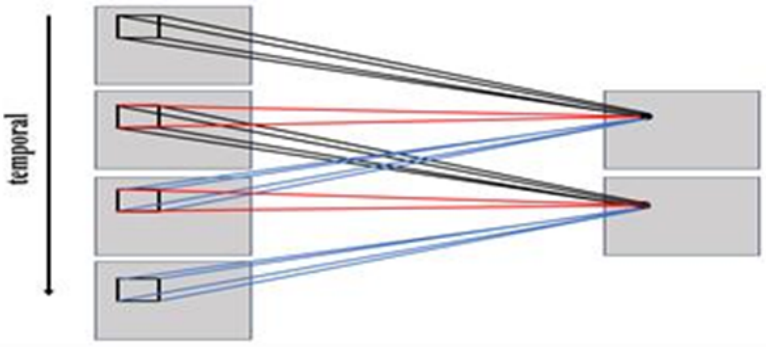

Fig. 3. 3D Convolution

2.2. Discussion on Target Detection and Human Skeletal Tracking Techniques

Target detection is a technology that identifies and locates specific objects or targets within images or videos. The successful application of deep learning, particularly convolutional neural networks (CNNs), has led to significant breakthroughs. The contem-porary methods of target detection are based on deep learning models, including the You Only Look Once (YOLO) [14] and Single Shot MultiBox Detector (SSD) algo-Rithms [15]. These

methods facilitate the rapid and precise recognition of targets. Themfield of human skeletal detection is concerned with the technology used to identify and track critical skeletal points of the human body in images or videos. There have been notable advances in this field in recent years due to the advent of deep learning- base methodologies. Several skeletal models, including OpenPose [16] and AlphaPose [17],have been used in various applications.

In the field of motion capture, various technologies such as OpenPose,DeepLabCut [18], and Kinect SDK [19] are widely used, but they have limitations in terms of computational resource requirements, applicability, and deployment flexibility. OpenPose offers high accuracy and multi-person pose estimation capabilities; however, its high computational cost makes real-time applications challenging. DeepLabCut is designed for biomedical applications and can improve recognition accuracy for specific subjects through transfer learning, yet it relies heavily on high-performance GPUs for training and inference, making it unsuitable for multi-object detection. Kinect SDK, on the other hand, depends on specialized hardware, limiting its adaptability in diverse environments. In contrast, MediaPipe [20] and OpenCV [21] provide superior efficiency and cross-platform compatibility, making them ideal for real-time motion capture applications. MediaPipe features a built-in Pose Landmarker, capable of running on both CPUs and GPUs, offering lightweight and efficient pose esti mation. OpenCV provides robust image processing capabilities for data preprocessing and feature extraction. Compared to other technologies, MediaPipe and OpenCV are more suitable for low-resource environments while ensuring stable real-time analysis, making them the preferred choice for this study.

This study selected OpenCV and MediaPipe as the primary tools for implementing the detection of the target above and the human skeletal detection methods. OpenCV provides a comprehensive image processing library, while MediaPipe offers efficient human skeletal detection capabilities. OpenCV (Open-Source Computer Vision Library) is a software library that assists developers in processing and analyzing various images and videos to perform multiple computer vision tasks. OpenCV has evolved significantly, becoming a cross-platform tool that supports many programming languages, including C++, Python, and Java. RGB image processing represents a fundamental function within the OpenCV framework. RGB stands for the red, green, and blue channels, and mixing these three colors at different intensities can represent a wide range of colors. In OpenCV, images are stored in matrix form, with the color information of each pixel represented by three matrices corresponding to the red, green, and blue channels. OpenCV splits an RGB image into three separate color channel matrices when reading it. These matrices contain the intensity values for each pixel in each channel. Once the image has been read, the intensity distribution can be observed by displaying the image for each channel. Separating the RGB channels of the image produces three separate greyscale photos, each representing the intensity of a one-color channel. In contrast, these three separate channels can also be combined into a single RGB image.

This study also used MediaPipe, an open-source visual computing framework developed by Google, used primarily for various vision-related tasks such as pose estimation,hand tracking, and face recognition. It uses CNNs to perform a detailed analysis of human images, accurately identifying key skeletal points. These models are trained on large datasets of images annotated with human skeletal points. With the optimizations and enhancements MediaPipe provides, they can efficiently and reliably perform full-body

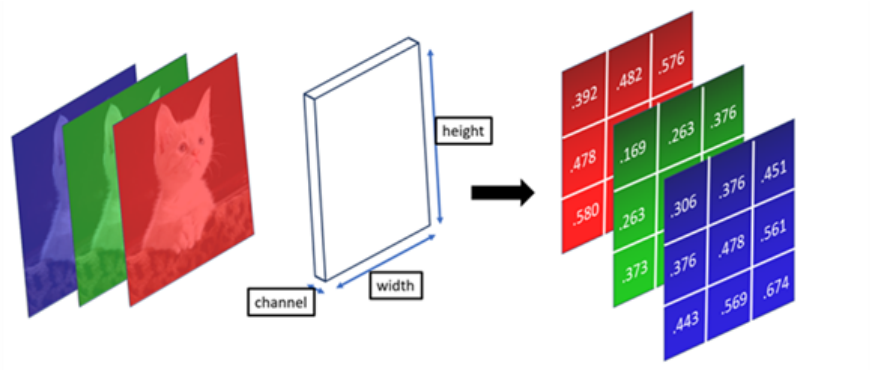

Fig. 4. OpenCV Processing Diagram

skeletal recognition in real-time video streams. The core concept of these techniques is to detect critical points on the human body (such as the head, shoulders, elbows, wrists, knees, and ankles) in images, thus accurately reconstructing human posture.

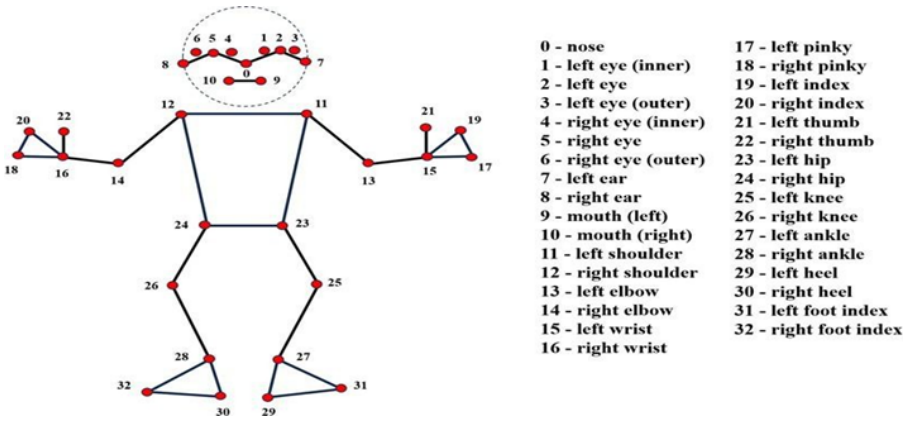

Fig. 5. MediaPipe Full-Body Skeletal Keypoints

3. Research Methodology

3.1. Research Process

Human figure recognition and the use of 2D skeletal information for feature correction and fitting are first discussed. It then explores how depth cameras can complement 3D human skeleton estimation and coordinate system transformation. The system's use of depth image information to obtain the critical points of the skeleton's third axis and to perform a

coordinate system transformation that facilitates the conversion of skeletal rotation values is explained. Next, we discuss the method for calculating the skeletal orientation. This involves using the 3D skeletal data to construct vectors and convert the skeletal rotation values. This step is critical for accurately describing the direction of human motion. Then, data transfer and numerical conversion are examined. The data transfer methods and numerical conversion techniques for skeletal data used in this study are described in detail, including how the calculated skeletal data is transferred to the Unreal Engine and how consistency and accuracy of the data are achieved. Finally, following the overall research process and framework, each step will be detailed, from data acquisition, processing, and skeletal estimation to the final presentation of the results. Each stage's functionality and synergistic interactions will be described to show this system's complete architecture and operational process.

3.2. 2D Human Skeleton Estimation and Feature Adjustment

The human location, skeletal positioning, and feature-matching process using technologies such as OpenCV and MediaPipe are detailed. The literature mentions using OpenCV's object recognition models to accurately identify individuals' position and appearance features and using MediaPipe's human skeletal models to locate human joint points accurately. Additionally, this study highlights three user-defined vital points added during the skeletal positioning process to enhance the convenience of subsequent computations and applications. In practice, the accuracy of depth information requires appropriate positional adjustments and calibrations. This includes maintaining

the optimal distance between the camera (Intel RealSense D435i) and the subject (approximately 2 meters), and making the necessary corrections and adjustments based on the actual environment.

During the measurement, the position and angle of the camera should be adjusted to minimize potential measurement errors. These adjustments help improve the depth of the information's accuracy and increase the data's reliability in application scenarios. To further improve the accuracy of depth information, it is recommended that multiple tests and calibrations are performed in different application environments to determine the optimum camera settings and operating methods. This approach effectively reduces errors caused by environmental variations, thus improving the overall stability and accuracy of the system. This process involves three main steps: object detection, skeleton positioning, and feature adjustment.

(1) Use OpenCV's object recognition models for human localization and identification.

(2) MediaPipe's human skeleton model positions the skeleton on the human image identified by OpenCV. Based on deep learning, MediaPipe's pose estimation model can accurately locate human joints from pictures and obtain the XY coordinates of each joint. The model adapts and adds three new vital points: the center of the shoulders, the center of the hips, and the center of the torso.

(3) Feature adjustment and correction for misidentified individuals are performed based on skeletal and joint position information. This includes eliminating or correcting potential errors in joint positioning, such as misidentifying parts of the scene or nonhuman targets as humans, which could lead to misapplication of the skeletal model and abnormal

values. These adjustments ensure that the information and feature values obtained more accurately reflect the proper posture and structure of the target individual.

3.3. Transformation of skeleton rotation value

In virtual character motion simulation, accurate and smooth rotation calculations are crucial for user experience. Euler angles, often used for rotation, can suffer from gimbal lock, which limits the system's ability to control rotation freely. To avoid this, quaternions are commonly used in fields like computer graphics, robotics, and motion capure, as they can represent rotations without gimbal lock and provide smoother interpolation [22]. A quaternion consists of a real part ω and three imaginary parts x, y, z, and can represent a 3D rotation. The quaternion is defined as:

$$q = \omega + xi + yj + zk \tag{2}$$

Where ω is the cosine of half the rotation angle and (x, y, z) are the components of the rotation axis multiplied by the sine of half the angle. To calculate the rotation of a skeleton, we first identify two points on a bone, say A and B, with initial coordinates. An array is used to record the three-dimensional coordinates of all skeletal points in this initial frame. The second identified frame is designated as the latest, and the 3D coordinates of all skeletal points are recorded in another array. The 3D coordinates identified in the second frame will continuously update this array until the program terminates. Given that the skeleton comprises numerous segments, each formed by two points, the direction of a bone segment following rotation can be determined by calculating the vectors formed by the two endpoints of the segment between the initial frame and the most recent frame. This process yields a series of vectors that represent the skeleton's direction of rotation. These vectors can then be used to calculate the skeleton's rotation, with the rotation values being represented as quaternions.

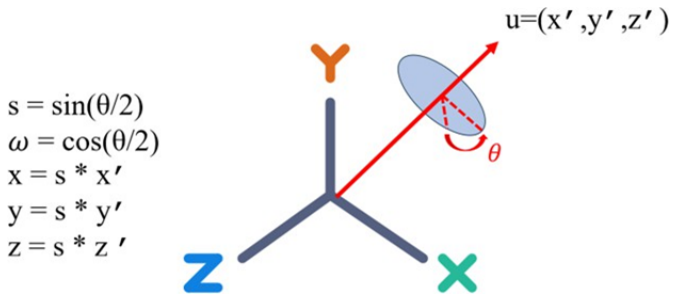

Fig. 6. Quaternion Illustration

The quaternion for this rotation is derived as:

$$q = (\cos\tfrac{\theta}{2}, \sin\tfrac{\theta}{2}\widehat{r}) \tag{3}$$

where [D835?][DC5F?]ˆ is the unit vector of the rotation axis, and θ is the angle between the vectors. If the norm of the cross product of the vectors is zero, it implies the vectors are parallel, and no rotation is needed. In such a case, the quaternion is set to [1, 0, 0, 0]. However, if the vectors are not parallel, the rotation axis is calculated as the unit vector of the cross product, and the rotation angle is the angle between the vectors. This information is used to generate the corresponding rotation quaternion. Through this process, we can accurately describe and apply the rotation from point A to point B in the motion reproduction of virtual characters, ensuring smooth and realistic motion without gimbal lock, and enhancing the overall animation quality [23].

3.4. Data Transmission and Numerical Conversion

It is of the utmost importance that this process undergoes rigorous data accuracy and transmission stability verification. It is of the utmost importance that this process be carried out to guarantee the correct application of external data within the Unreal Engine. This process enables the accurate tracking and presentation of the skeletal direction and coordinates. The precision and fluidity with which virtual characters can perform movements depend on the data's accuracy and the reliability of their transmission. This ensures that users will experience enhanced interaction with the virtual environment. Once a successful socket connection has been established, the Received Message event will initiate the reception of the overall skeletal rotation values transmitted from Python. Rotation values are associated with nine distinct segments of the skeletal system. The data related to each segment comprises seven values: the world coordinates (X, Y, Z) and the rotation quaternion (X, Y, Z, W). The primary rationale for selecting

the quaternion method is that it offers a straightforward and accurate approach to handling rotational data. Although traditional Euler angles are intuitively appealing, they are susceptible to gimbal lock issues during calculations, which increases the complexity of data processing. On the contrary, quaternions circumvent this issue, offering a stable and accurate description of rotations, dealing with rotational data. Although traditional Euler angles are intuitively appealing, they are susceptible to gimbal lock issues during calculations, which increases the complexity of data processing. On the contrary, quaternions circumvent this issue, offering a stable and accurate description of rotations.

The diagram below illustrates how the first three Read float nodes are employed to receive the world coordinates of the skeletal points. The coordinate data represent the positions of the skeletal rotation points in three-dimensional world space. Subsequently, the data from the aforementioned skeletal rotation points is transmitted to the Make Vector node, which is incorporated into the location array. This approach effectively manages and stores all the skeletal rotation points' position coordinates. The final four Read float nodes receive the quaternions of the skeletal rotation points. The quaternions thus describe the rotational direction of each skeletal rotation point, representing its rotational state in the world coordinates. Subsequently, the quaternions of these skeletal rotation points are transmitted to the Make Quat node and incorporated into the quaternion array, ensuring accurate storage and utilization of the quaternion data. Subsequently, the quaternions are converted into the rotation type. This design enhances the efficiency and accuracy of data processing and reduces the need for data conversion in subsequent applications. Finally, the Received Message event is linked to nine quaternion methods, thus completing the

data processing. This process ensures that all data points are correctly received and processed, guaranteeing the system's stable operation and high performance.

To align the spatial coordinates identified by the camera with those of the Unreal Engine, this study devised a rudimentary human skeleton within the scene (level). The skeletal information from the TCP socket blueprint is called the Level Blueprint, with the location and rotation information being passed sequentially to each skeleton joint [24]. This ensures the precise synchronization of the data. Once the initial configuration has been completed, the skeletal data will be imported into the relevant pins. This process enables the data captured by the camera to be transmitted to the corresponding skeleton within Unreal Engine [25]. Subsequently, the skeletal data are transmitted to the scene, where they are converted to the skeleton, as illustrated in the schematic diagram. Despite the use of world coordinates by both the camera and the scene in Unreal, an offset exists due to differing origins. Consequently, direct use of the coordinates captured by the camera will result in positional errors. To guarantee the synchronization of coordinates, it is necessary to incorporate an offset into the location data before its transmission to the skeleton. This step aims to correct the origin offset, ensuring data consistency and accuracy. Consequently, the discrepancies above are eliminated, thus eliminating errors caused by differences in the origins of the coordinate system.

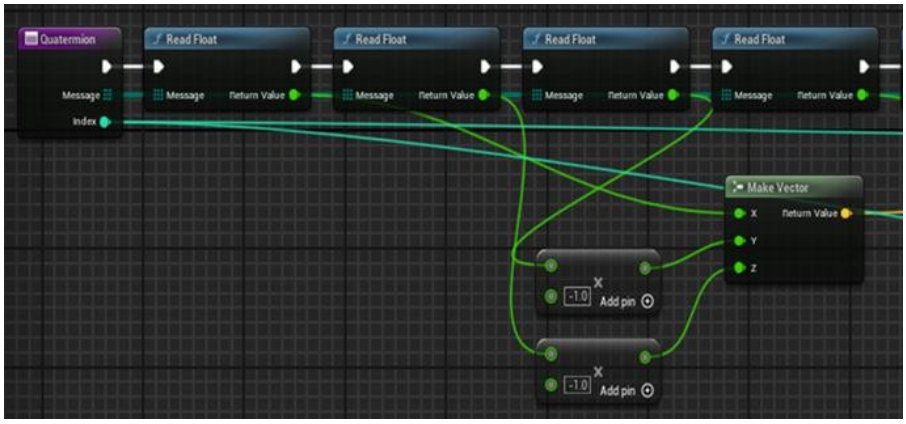

Fig. 7. Socket Connection Event Blueprint 1

4. Research Results and Discussion

The approach proposed by Lao et al. [26], which centers around virtual characters in theatrical performances, aims to enhance learners' language abilities and stimulate their interest in learning. The value of this method lies in its ability to help learners better understand the language through realistic performances and interactions with virtual characters, while also improving learning outcomes via immediate feedback and personalized interaction. The effectiveness of virtual characters in language learning has been demonstrated

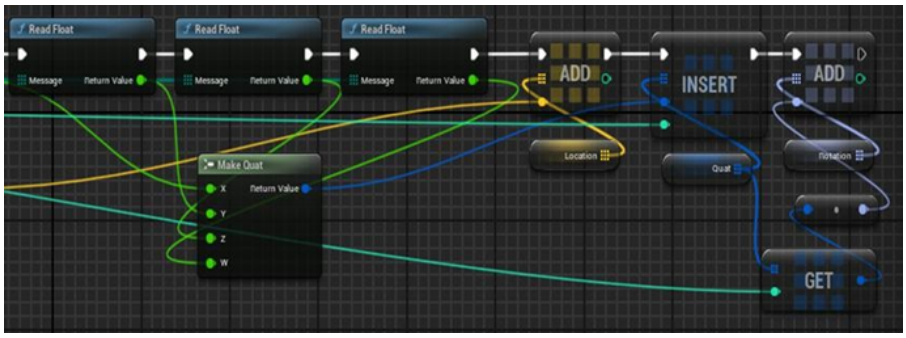

Fig. 8. Socket Connection Event Blueprint 2

in numerous studies. This approach not only simulates real-life communication scenarios, allowing learners to practice in a safe environment, but also creates an interactive learning atmosphere, further increasing learners' engagement and motivation.

In this study, six typical movements of virtual characters (such as kicking left, punching right, squatting, etc.) were selected as research subjects. These movements represent fundamental human motions and are frequently used in virtual character performances, making them highly relevant and practical. These movements were chosen because they effectively test the technology of converting human skeleton coordinates into digital avatars and ensuring that the virtual character's motions accurately reflect the user's actual movements HTML [27]. The experiment used a self-developed system to record the user's motion data in real-time and convert this data into the movements of a digital avatar in Unreal. The goal was to verify the accuracy of the technology that converts human skeleton coordinates and synchronizes movements in real-time. The system design is based on previous research, such as open-source skeleton tracking and real-time animation synchronization technologies, which have been shown to effectively capture and reproduce motions. Recording videos during the experiment helps with subsequent analysis and comparison, providing rich data support to ensure the accuracy of the technology and facilitate continuous improvements.

The analysis of the videos showed that the system could accurately reproduce the user's movements with high precision and low latency, confirming the effectiveness and feasibility of the proposed method [28]. This not only achieves ideal results in the performance of the virtual character's movements but also enhances the interactive experience for the learner. In conclusion, virtual characters have great potential in language learning and other application scenarios. These technologies are expected to be further applied in various fields, bringing innovative changes to education, entertainment, and social industries. To verify this study's primary objective, the system's accuracy in recognizing six specific actions, left kick, right kick, left punch, right punch, squatting, and sitting, will be evaluated using a confusion matrix as the primary analysis tool and multiple tests. The objective is to obtain accurate data to improve the reliability of the test results.

4.1. Experimental Environment

The experimental environment for the system in this study begins with the configuration and activation of the server side to ensure that the server runs correctly. Subsequently, the user establishes a connection by inputting the server's fixed IP address, ensuring stable data transmission from the server to the user side. The server is primarily responsible for processing and managing the data in this process. The server receives data from input devices, performs the necessary data processing and analysis, and then transmits the processed data to the user side. It is paramount that the server is capable of efficient processing and stability, as these are the two key factors that ensure that the system functions correctly.

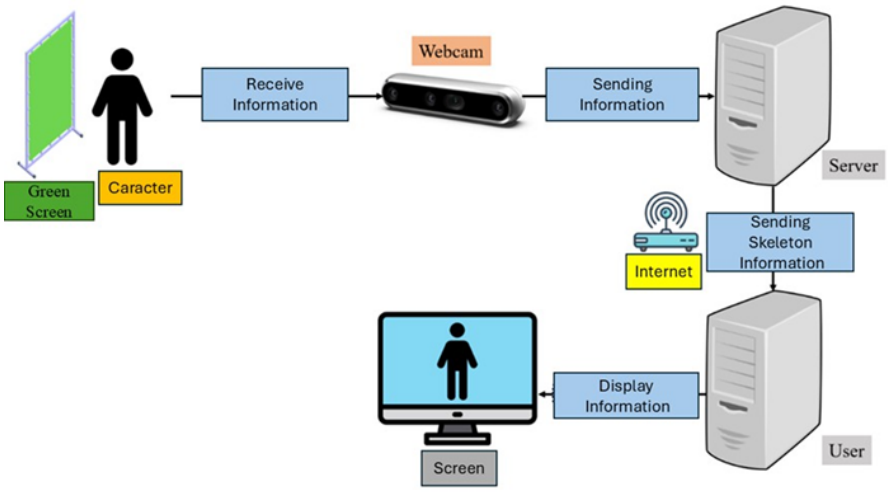

Fig. 9. Schematic Diagram of the Experimental Environment

The user interface connects to the server, which transmits the required data. Subsequently, the data is employed to simulate and regulate the virtual character's movements. It is of the utmost importance that the user interface can receive data from the server in real-time, as this data is used to simulate the virtual character's movements to ensure that the character's actions are accurately and smoothly reproduced. It is of the utmost importance that the connection between the server and the user side is stable and reliable, as this is a prerequisite for the system to function optimally. Any instability in the connection could result in interruptions or delays in data transmission, consequently affecting the performance of the virtual character's movements. It can be reasonably concluded that the server and user environments must exhibit high stability and speed to ensure optimal performance.

4.2. Experimental Evaluation Metrics

This study employs the confusion matrix as the principal metric for experimental evaluation to quantify the system's performance in capturing and synchronizing human skeletons in virtual space. The confusion matrix is a widely employed methodology for assessing the efficacy of classification models, offering a range of metrics, including precision and recall, to comprehensively evaluate the system's performance. The confusion matrix is a specific tabular structure used to describe the performance of a classification model in classification tasks. The model's classification results are presented by comparing the actual and predicted classifications.

True Positive (TP): The number of instances where the actual action is performed, and the Unreal model correctly displays the action.

True Negative (TN): The number of instances where the actual action is not performed, and the Unreal model correctly displays that the action is not performed.

False Positive (FP): The number of instances where the actual action is not performed, but the Unreal model incorrectly displays the action as being performed.

False Negative (FN): The number of instances where the actual action is performed, but the Unreal model incorrectly displays that the action is not performed. The confusion matrix enables the determination of the system's performance, which is evaluated using the following metrics.

Accuracy: This represents the proportion of correct predictions made by the model and is an essential indicator of overall performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Precision: represents the proportion of samples predicted to be a specific action that is an action, reflecting the reliability of the model's predictions for that action.

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

Recall: represents the proportion of samples with a specific action correctly predicted as that action, reflecting the model's sensitivity.

$$Recall = \frac{TP}{TP+FP} \tag{6}$$

Next, the system's accuracy in recognizing six specific actions described in the literature is analyzed. These actions include kicking left, kicking right, punching left, punching right, crouching, and sitting. The analysis will determine whether these actions are accurately rendered in Unreal Engine.

4.3. Experimental Testing and Analysis

This study focuses on the system's accuracy in recognizing six types of action out of 100 instances. We will calculate and analyze various metrics in detail, including accuracy, precision, and recall, to evaluate the system's recognition performance. These metrics will demonstrate the system's accuracy in recognizing these six actions and reveal any

potential recognition errors, providing a basis for subsequent system improvements. The analysis starts with 100 instances of video recording and analysis of

27 the right-kicking action (followed by the other five actions). Figure 9 shows the correct identification of the right kick action, while another figure shows the incorrect identification of the same action.

Fig. 10. Schematic Diagram of the Experimental Environment

In this analysis of action recognition, we found that the system made some errors in identifying the action kick right. Out of 100 instances, the system correctly identified kick right 45 times but made five incorrect identifications. Additionally, 50 actions were not kick right, and the system correctly identified all 50 actions as not kick right. Therefore, the number of non-kick right actions misidentified as kick right is zero. Below is the confusion matrix for this action analysis and the calculation results for the confusion matrix of the kick right action.

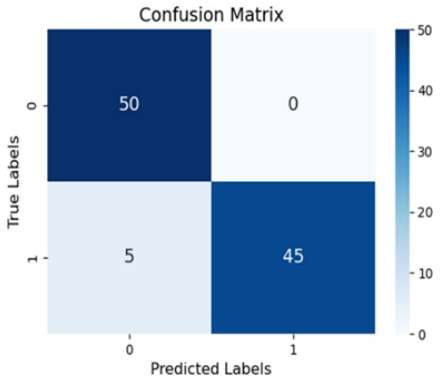

Fig. 11. Kick Right Confusion Matrix and Calculation Results

4.4. Experimental Testing and Analysis

To evaluate the system's accuracy in recognizing six actions, we analyzed experimental data from 100 instances. We calculate each action's accuracy, precision, and recall to evaluate the system's recognition performance. As shown in Table 1, the experimental results indicate that although the system performs well in recognizing most actions there is still room for improvement.

Table 1. Results of the experimental analysis of the confusion matrix

	Precision (%)	Accuracy(%)	Recall(%)
Kick Left	96	100	92
Kick Right	95	100	90
Punch Left	97	100	94
Punch Right	95	100	90
Squat	94	100	88
Sit	98	100	96
Overall Average	95.84	100	91.67

We can draw the following conclusions based on a comprehensive analysis of the above results.

High accuracy: The system achieved a 100% accuracy rate for all actions, which was almost always correct once the system identified an action. This reflects the system's strong ability to avoid false positives.

Insufficient recall: Despite the high accuracy, the recall rate was relatively low, especially for certain actions such as 'crouching' and 'kicking right,' with recall rates of 88% and 90%, respectively. This indicates the need for further adjustments to improve the system's recall, demonstrating our commitment to its ongoing development.

Overall high accuracy: The average accuracy rate for all actions was more than 94%,

demonstrating the overall stable recognition performance of the system. However, the accuracy of individual actions must be improved to achieve a more comprehensive recognition capability.

5. Conclusions

The system developed in this study demonstrated a precision rate of 100% for all action recognitions. This indicates that it was almost always correct once the system identified an action. This reflects the system's robust capacity to avoid false positives, particularly in recognizing everyday actions such as walking, running, and waving. The system demonstrated the ability to identify these actions with a low incidence of misclassification accurately.

Despite the high precision, the recall rate was relatively low, particularly for specific actions such as crouching and kicking right, with 88% and 90% recall rates, respectively. This indicates that the system occasionally fails to identify these actions correctly. Further optimization may be required for these actions' feature extraction and recognition models.

For instance, the system continues encountering difficulties recognizing complex and fast actions. This requires the acquisition of additional training data and the implementation of optimized algorithms to enhance recall rates. The system demonstrated an accuracy rate of more than 94% for all actions, indicating stable and reliable recognition performance. This suggests that the system performed consistently and reliably in recognizing most actions, whether simple or complex sequential movements, maintaining a high accuracy rate. However, further enhancements are necessary to achieve a more comprehensive recognition performance, particularly with respect to the accuracy of specific individual actions.

This study's experimental results demonstrate the system's potential in action recogition while identifying specific areas for improvement. Collective data indicate that the system achieved a precision rate of 100%, an accuracy rate of 95.84%, and a percent recall rate of 91.67%. These figures demonstrate that the system has a high recognition capability, yet there is room for improvement. Further research can be conducted to build upon these findings to optimize the system's accuracy and stability in recognizing various actions, thereby enhancing its applicability in real-world scenarios. In future experimental procedures, individual differences in participants' physical characteristics will be progressively considered, and environmental variables during the experiments—such as temperature, clothing effects, and body shape—will be controlled to maintain the system's accuracy at a consistent level. Additionally, in terms of future research directions, building on the system's current foundation, we aim to use the recognition of these six basic movements as a basis to develop more refined motion representations in the next phase. As the number of recognized basic movements increases, it may become feasible to apply the system to recognize sequences of continuous actions—such as Tai Chi or wellness exercises—thereby adapting the system to a wider range of application scenarios.

References

1. Sathe, P.S., *Tracking, Recognizing and Analyzing Human Exercise Activity*. University of Akron.(2019)
2. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, **25**.(2012)
3. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* nature, **323**(6088): p. 533–536.(1986)
4. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural computation. **9**(8): p. 1735–1780.(1997)
5. Cho, K., et al., *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* arXiv preprint arXiv:1406.1078.(2014)
6. Tran, D., et al. *Learning spatiotemporal features with 3d convolutional networks*. in *Proceedings of the IEEE international conference on computer vision*. (2015)
7. Carreira, J. and A. Zisserman. *Quo vadis, action recognition? a new model and the kinetics dataset*. in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2017)
8. Qiu, Z., T. Yao, and T. Mei. *Learning spatio-temporal representation with pseudo-3d residual networks*. in *proceedings of the IEEE International Conference on Computer Vision*. (2017)
9. Xie, S., et al. *Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification*. in *Proceedings of the European conference on computer vision (ECCV)* (2018)

10. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2015)
11. Zhou, B., et al. *Temporal relational reasoning in videos*. in *Proceedings of the European conference on computer vision (ECCV)*.(2018)
12. Zolfaghari, M., K. Singh, and T. Brox. *Eco: Efficient convolutional network for online video understanding*. in *Proceedings of the European conference on computer vision (ECCV)*.(2018)
13. Wang, L., et al. *Temporal segment networks: Towards good practices for deep action recognition*. in *European conference on computer vision*. Springer.(2016)
14. Bochkovskiy, A., C.-Y. Wang, and H.-Y.M. Liao, *Yolov4: Optimal speed and accuracy of object detection.* arXiv preprint arXiv:2004.10934.(2020)
15. Liu, W., et al. *Ssd: Single shot multibox detector*. in *European conference on computer vision*. Springer.(2016)
16. Cao, Z., et al. *Realtime multi-person 2d pose estimation using part affinity fields*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2017)
17. Fang, H.-S., et al., *Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time.* IEEE transactions on pattern analysis and machine intelligence, **45**(6): p. 7157–7173.(2022)
18. Mathis, A., et al., *DeepLabCut: markerless pose estimation of user-defined body parts with deep learning.* Nature neuroscience, **21**(9): p. 1281–1289.(2018)
19. Izadi, S., et al. *Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera*. in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. (2011)
20. Lugaresi, C., et al., *Mediapipe: A framework for building perception pipelines.* arXiv preprint arXiv:1906.08172, (2019)
21. Bradski, G. and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.(2008)
22. Shoemake, K. *Animating rotation with quaternion curves*. in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*.(1985)
23. Alvarado, E., D. Rohmer, and M.P. Cani. *Generating Upper-Body Motion for Real-Time Characters Making their Way through Dynamic Environments*. in *Computer Graphics Forum*. Wiley Online Library.(2022)
24. Edeline, K., et al., *Using UDP for internet transport evolution.* arXiv preprint arXiv:1612.07816. (2016)
25. Qiu, W., et al. *Unrealcv: Virtual worlds for computer vision*. in *Proceedings of the 25th ACM international conference on Multimedia*. (2017)
26. Huang, X., et al., *A systematic review of AR and VR enhanced language learning.* Sustainability, **13**(9): p. 4639.(2021)
27. Younes, M., *Learning and simulation of sport strategies (boxing) for virtual reality training*, Université de Rennes.(2024)
28. Yan, Z. and J. Yi, *Dissecting Latency in 360 Video Camera Sensing Systems.* Sensors, **22**(16): p. 6001.(2022)

Fei-lung Lin, he was born and raised in Taiwan. Currently studying at the Institute of Technological and Vocational Education, National Taipei University of Technology, Taiwan. A strong personal interest in science, technology, and educational environments drives my pursuit of academic research.

Jui-Hung Kao, is an associate professor at Shih Hsin University since 2023. During his tenure as project manager at the Research Center for Humanities and Social Sciences in

2014, he was responsible for the administrative business of research and program execution, which combined statistical methods with spatial information visualization and is good at writing programs and data analysis. The empirical research topics focus on three parts: spatial data analysis, medical management research, and long-term medical poicy.

Yu-Yu Yen, she has been working as a lecturer in the Center for General Education at Shih Hsih University since 2022.She is also currently enrolled in a PhD program in the Department of Biomedical Engineering, National Yang Ming Chiao Tung University.

Kuan-Wen Liao, was born and raised in Taiwan; a fervent interest in the intersection of technology and management has driven my academic and professional pursuits. In July 2024, I was honored to receive my Master's degree in Information Management from the distinguished Shih Hsin University.

Pu Huang, has been a faculty member at Shaoguan University since September 2020. His research fields such as administrative management, social governance, and public policy analysis.

Received: October 02, 2024; Accepted: June 27, 2025.