ISSN:1738-6764, eISSN:2093-7504

Posture Recognition for Human Interface in Virtual Reality

Eun Young Ahn [1,*]

[1] Dept. of Intelligence Media Engineering, Hanbat National University; aey@hanbat.ac.kr
* Correspondence

https://doi.org/10.5392/IJoC.2026.22.1.030

Manuscript Received 24 July 2025; Received 5 December 2025; Accepted 16 December 2025

Abstract: *This paper introduces a method for improving the accuracy of human posture recognition using non-wearable sensors. We tackle the challenges associated with motion recognition when using a low-precision depth camera. In our study, we define pose recognition as a classification task within a multi-dimensional space. We propose a spatial modeling approach that utilizes a lazy learning system, enabling robust posture recognition despite low-quality motion data or significant variations in user actions. The input motion data is transformed into informative feature vectors, normalized to the user's initial pose. Experimental results show high performance and accuracy, even in the presence of unstable or erroneous motion input.*

Keywords: Posture Recognition; Disparity Vector; Human Action; Depth Motion Sensor; Feature Vector

1. Introduction

Human motion recognition has become increasingly vital across various domains, including Human-Computer Interaction (HCI), gaming, and video surveillance. Despite decades of extensive research, it remains a formidable challenge. In particular, motion recognition from monocular video streams suffers from inherent performance limitations. Over the past decade, however, the emergence of cost-effective depth cameras has shifted the paradigm of human activity recognition. Many contemporary studies exploit both depth and skeleton data to achieve high accuracy, with state-of-the-art approaches typically utilizing deep learning models such as Spatio-Temporal Graph Convolutional Networks (ST-GCN) or Transformers [1-4]. While effective, these methods often require large-scale annotated datasets and substantial computational resources (e.g., high-end GPUs), posing significant barriers for real-time applications on low-end consumer PCs or standalone VR environments. Furthermore, depth estimation remains susceptible to noise and fluctuating background conditions. Skeleton-based approaches, which utilize joint locations extracted from depth images [5], have seen various spatiotemporal attempts at action recognition [6-8]. However, treating motion as a continuous sequence is not always optimal for real-time VR interaction. In VR applications, the immediate recognition of a user's posture at a specific moment is often more critical than the analysis of long temporal sequences.

A primary challenge in this domain is the inherent noise in data from low-cost sensors, caused by resolution limits, jitter, and body occlusion. Additionally, variability in how users perform the same pose—due to physical differences or skill levels—further complicates recognition. To address these challenges without the heavy computational overhead of deep learning, we propose an informative representation termed the "Feature Vector" coupled with a lazy learning classification algorithm. This paper presents a method to enhance real-time human posture recognition accuracy using non-wearable sensors, specifically within the context of VR Skydiving. To this end, we define and recognize a set of standard maneuvers essential for a functional skydiving interface.

2. Related Works and Challenging Issues

2.1 Limitations of Existing Approaches

Human action recognition has been extensively studied using various modalities. Early approaches focused on RGB video analysis, but the advent of depth cameras shifted the focus toward skeletal data [5]. Recent state-

of-the-art methods largely rely on deep learning architectures. For instance, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are commonly used to model temporal dependencies in motion sequences [9]. More recently, Graph Convolutional Networks (GCNs), such as ST-GCN, have demonstrated superior performance by modeling the human body as a graph structure [1-4]. However, these deep learning-based methods typically require high computational resources (e.g., GPUs) and induce processing latency, making them unsuitable for real-time interaction on standalone VR headsets or low-spec consumer PCs. Furthermore, they are often trained on large-scale datasets for general actions (e.g., walking, waving), which do not align with the specific, fine-grained maneuvers required for applications like VR skydiving.

2.2 Challenging Issues in Depth-based Recognition

While depth cameras provide feasible quality under optimal conditions, data from these sensors are inherently imprecise. Extracted skeletal data often exhibit significant jittering within a wide range. Figure 1-(a) illustrates the plotting results of thousands of skeletal data points captured from a single subject performing a specific pose over 35 seconds. This demonstrates the typical inaccuracy of input data when using consumer-grade RGBD cameras. Specifically, variations in joint positions occur even for the same pose. The angular difference between the same joints in a sequence can reach up to 15°, which poses a significant challenge for accurate pose recognition.

Although the Field of View (FOV) and sensing range of these cameras have improved, parts of the human body often move outside the camera's effective range during full-body motion. When the depth data includes occluded or out-of-range body parts, it becomes difficult to estimate joint positions, leading to unstable and error-prone data, as shown in Figure 1-(b). Body occlusion is a critical issue that causes severe estimation errors and reduces recognition accuracy.

Another complicating factor is inter-user variability. Users typically perform the same pose slightly differently depending on their physical attributes, skill level, or experience. This variability presents a significant challenge for applications that utilize similar but distinct maneuvers as a user interface.

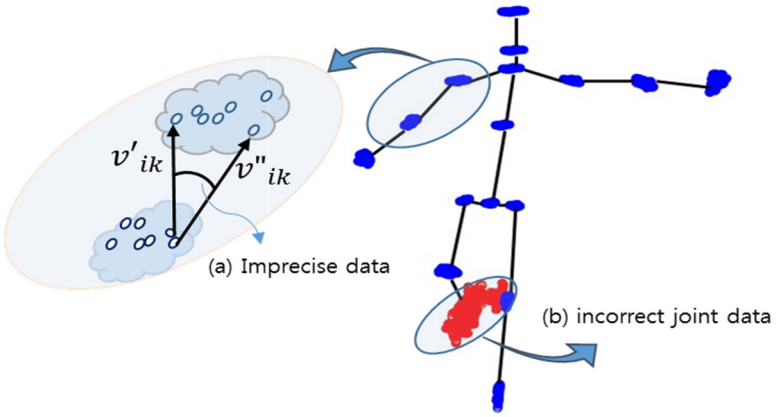


Figure 1. Characteristics of skeletal data from depth cameras: (a) Imprecise joint data showing angular variations over time for a static pose; (b) Incorrect and unstable joint estimation resulting from body occlusion.

3. The Proposed Method

The objective of this research is to develop a posture recognition system that functions as a non-wearable user interface in a real-time VR environment. We formulate posture recognition as a classification task within a multi-dimensional feature space. Robust feature extraction is essential to address the inherent variability in user body shapes and the error-prone nature of estimated skeletal data. To this end, we propose a method to extract invariant features that reliably discriminate postures regardless of individual action patterns or imprecise skeletal inputs.

3.1 Basic Action Vectors

Skeletal joint data, when represented by absolute coordinates, are subject to significant variation depending on the subject's distance from the camera and their body proportions. To mitigate the depth

quantization errors discussed in Section 2 and ensure invariance to camera distance, we utilize relative joint vectors instead of raw position data.

As illustrated in Figure 2, a joint vector v_k is defined by the spatial relationship between a joint J_k and its adjacent neighbor joint $J_{neighbor}$. This relationship is expressed in Equation (1):

$$v_k = J_k - J_{neighbor} \quad (1)$$

Where J_k and $J_{neighbor}$ represent the 3D coordinates (x, y, z) of the respective joints. This vector representation captures the intrinsic geometric relationships of body parts, providing a more stable and robust input compared to absolute spatial coordinates.

To eliminate the influence of a subject's physical proportions, each joint vector is normalized by its magnitude to obtain a unit vector $\hat{v}_k$. This process ensures that the system focuses solely on the directional component of the posture, regardless of the user's bone length. The normalization is defined as:

$$\hat{v}_k = \frac{v_k}{\|v_k\|} \quad (2)$$

where $\|v_k\|$ denotes the Euclidean norm of the vector. By utilizing unit vectors, the feature space becomes invariant to the physical scale of the user, allowing for a more generalized classification.

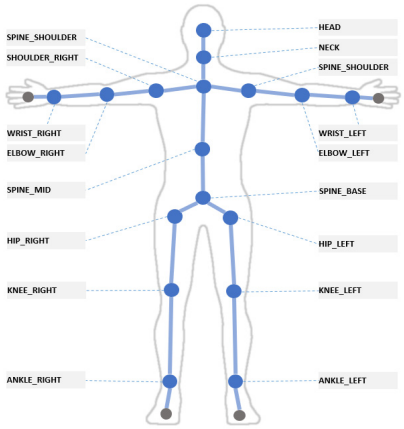


Figure 2. The human skeletal joint structure for posture recognition

3.2. User-Invariant Representation

To achieve high recognition accuracy independent of the user's physical proportions (e.g., limb lengths), we introduce a Feature Vector composed of 16 difference vectors.

3.2.1 Difference Vector Calculation

The difference vector (d_{ik}) represents the spatial variation between the joint vector of the current pose (v_{ik}) and corresponding joint vector from the user's initial T-pose (v_{0k}), By using the T-pose as a personalized baseline, we can effectively normalize the skeletal data for each individual. This calculation is defined in Equation (3):

$$d_{ik} = v_{ik} - v_{0k}, k = 1,2,\ldots,N-1. \quad (3)$$

This subtraction process cancels out individual body constraints, such as specific bone lengths or postural habits. Consequently, it maps the same motion to a consistent region within the feature space, ensuring that the same action results in similar vectors regardless of the user's physical build.

3.2.2 Final Feature Vector Construction

The final Feature Vector (f_i) for a specific posture i is constructed by concatenating the set of 16 difference vectors. This high-dimensional vector serves as the standardized input for the classification system, defined in Equation (4):

$$f_i = [d_{i,0}, d_{i,1}, \ldots, d_{i,N-1}]^T, \quad i = 1, 2, \ldots, p \tag{4}$$

where N is the number of joint vectors (in this case, 16) and p is the total number of recognized posture classes. This representation effectively encodes the relative displacement of the body from the neutral T-pose, providing a robust and invariant feature set for real-time VR interaction.

Table 1. Differences of joints between a pose and basic pose

	$d_{i,0}$	$d_{i,1}$	$d_{i,2}$	$d_{i,3}$	$d_{i,4}$	$d_{i,5}$	$d_{i,6}$	$d_{i,7}$	$d_{i,8}$	$d_{i,9}$	$d_{i,10}$	$d_{i,11}$	$d_{i,12}$	$d_{i,13}$	$d_{i,14}$	$d_{i,15}$
Pose_0	-0.01	-0.1	-0.2	0.03	0.17	1.29	0.11	0.17	1.15	0.09	0.03	0.02	0.07	0.12	0.11	0.01
Pose_1	-0.12	-0.1	-0.06	0.02	0.25	1.20	-0.17	-0.26	-1.18	-0.05	0.33	0.14	0.08	0.08	-0.12	-0.16
Pose_2	-0.22	-0.05	-0.06	-0.02	0.00	0.00	0.11	-0.39	-0.41	-0.05	0.33	0.14	0.08	0.08	1.45	-0.16
Pose_3	-0.17	-0.04	-0.14	0.22	0.48	0.43	-0.14	0.00	0.00	0.05	0.39	1.35	-0.06	-0.15	-0.26	-0.13
Pose_4	-0.20	-0.09	-0.15	0.02	0.35	0.31	0.16	0.09	0.11	-0.05	0.31	0.17	0.08	0.37	1.19	-0.29
Pose_5	-0.18	-0.05	-0.05	0.19	0.19	0.18	-0.09	-0.28	-0.29	0.04	0.46	1.30	-0.04	-0.15	-0.35	-0.14
Pose_6	0.02	0.00	-0.05	-0.10	-0.88	1.12	-0.19	-0.99	-1.26	-0.01	0.04	0.11	0.01	-0.19	-0.04	0.01

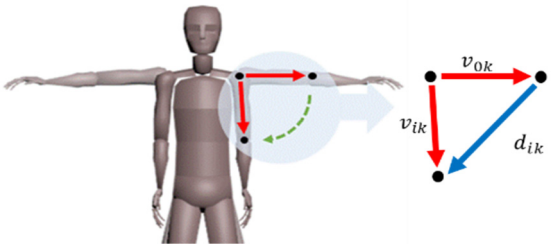


Figure 3. An example of the difference vector for a joint vector

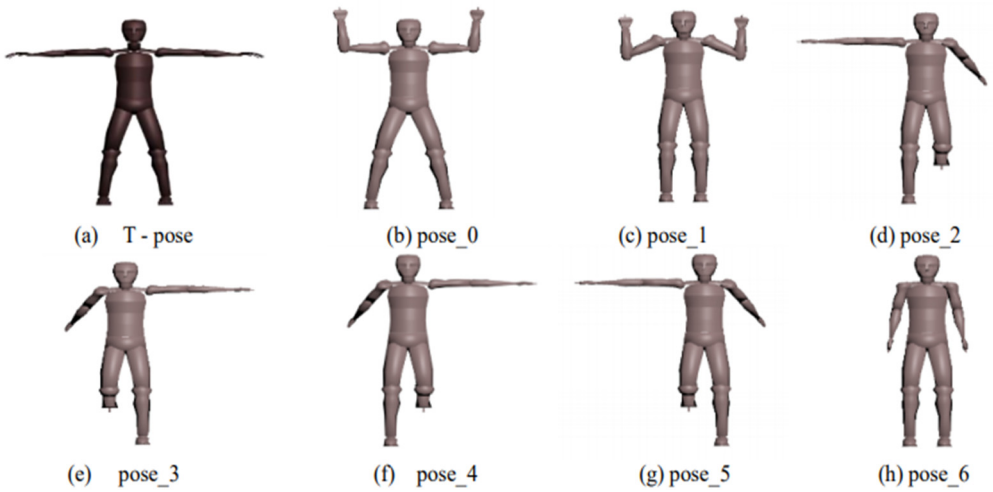
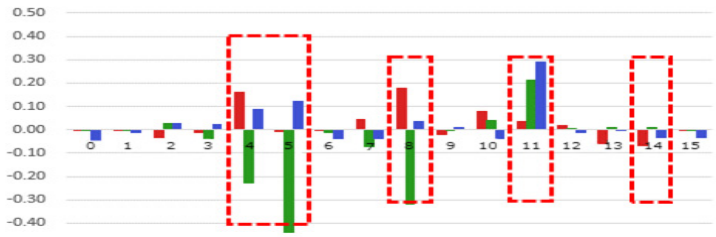


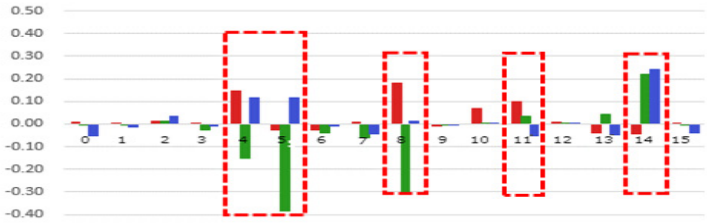
Figure 4. The Essential postures for VR Skydiving

Figure 4 illustrates example skydiving postures, where T-pose is designated as base pose. As shown in the figure, Pose 1 shares a structural similarity with Pose 0, differing only in the extent of the shoulder and leg angles. Similarly, Pose 2 and Pose 3 exhibit nearly identical configurations, with the primary distinction being which leg is raised (right vs. left).

Figure 5 presents the three-axial difference of each joint for the two similar postures, Pose 2 and Pose 3. Although several joint vectors such as $v5$, $v6$ and $v10$ fall within similar value ranges, the two poses are accurately distinguished by the distinct patterns observed in $v14$, $v18$. This demonstrates that the proposed feature vector provides a significant discriminant pattern for posture classification, even when the differences between postures are subtle. Consequently, the feature vector effectively captures fine-grained spatial variations that are essential for high-precision recognition in a VR skydiving interface.



(a) The differences of joints for pose 3



(b) The differences of joints for pose 4

Figure 5. Axial disparities of joints compared with pose T

3.3 Posture Classification and Stability Control

The extracted feature vectors are represented in a 48-dimensional space (16 vectors × axes). To classify these vectors in real-time, we employ a lazy learning approach based on the k-Nearest Neighbor (k-NN) algorithm. Unlike deep learning models that require extensive offline training and significant computational resources, k-NN allows for immediate adaptation to new data and maintains low computational overhead, which is ideal for standalone VR environments.

3.3.1 Similarity Measure: Pearson Correlation

Standard Euclidean distance is often sensitive to the absolute magnitude of vectors, making it vulnerable to sensor outliers and jittering. To prioritize the structural pattern (the "shape" of the pose) over absolute magnitude, we utilize Pearson Correlation as the similarity metric. The correlation coefficient $\boldsymbol{r}$ is calculated as shown in Equation (5):

$$r = \frac{\sum(f_i - \bar{f})(g_i - \bar{g})}{\sqrt{\sum(f_i - \bar{f})^2 \sum(g_i - \bar{g})^2}} \tag{5}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ represent the input feature vector and the template vector, respectively. This metric evaluates the linear correlation between vector patterns, providing robustness against sensor noise and varying intensities of user movements.

3.3.2 Transition State Management: Postponement Strategy

A critical challenge in continuous motion recognition is the "Transition State," which occurs when a user moves between two defined postures. During this phase, the feature vector is often ambiguous, leading to classification "flickering" (rapidly switching between classes). To address this, we propose a Postponement Strategy acting as a temporal filter. We introduce the Components Ratio Cr, defined as the proportion of the dominant class among the $\boldsymbol{k}$ nearest neighbors. The recognition decision is made only if the correlation distance is below a threshold and the Components Ratio exceeds a certainty level (e.g., 80%). If these conditions are not

satisfied, the system identifies the frame as a "Transition/Unstable" state. In this case, the system postpones the new decision and retains the previous stable state. This mechanism ensures that the VR interface reacts only to intentional and stable maneuvers, significantly improving the user experience by eliminating jittery transitions.

4. Experimental Results

4.1 Dataset and Experimental Environment

The tested dataset is specifically constructed for Virtual Reality Skydiving. The 7 selected poses (excluding the T-pose) correspond to the standard maneuvers defined in professional indoor skydiving training curricula, such as the "Standard Arch," "Left/Right Turns," and "Tracking." This domain-specific dataset is essential to validate the precise control interface required for the VR. To verify the performance of the proposed method, we conducted experiments on eight subjects with varying body conditions (height: 153cm–181cm, various body types) as shown in Table 2. Motion data was captured using a Microsoft Kinect v2 depth camera at 30 fps. A total of 56,000 frames were collected, with approximately 35 seconds per pose for each subject. simulator. We evaluated the proposed system using a Leave-One-Person-Out cross-validation scheme. Seven subjects participated in the learning process (constructing the feature space), and the remaining one subject was used for testing. This process was repeated for all subjects to test the system's invariance to new users. Learning data dictates the performance of recognition. Therefore, we removed and learned data with ambiguous distinctions between pose 0 and pose 1, which are similar poses, from the data used for training.

Table 2. The body condition of the eight subjects

Person ID	Height (cm)	sex	Body type	Captured data (7 poses)
A	166	M	big.	1,000 Frame * 7 Pose
B	177	M	normal	1,000 Frame * 7 Pose
C	172	M	big	1,000 Frame * 7 Pose
D	173	M	normal	1,000 Frame * 7 Pose
E	181	M	thin	1,000 Frame * 7 Pose
F	169	M	thin	1,000 Frame * 7 Pose
G	153	F	big	1,000 Frame * 7 Pose
H	172	M	normal	1,000 Frame * 7 Pose
Total				**56,000 Frame**

Table 3. The recognition results for the data include the ambiguous pose

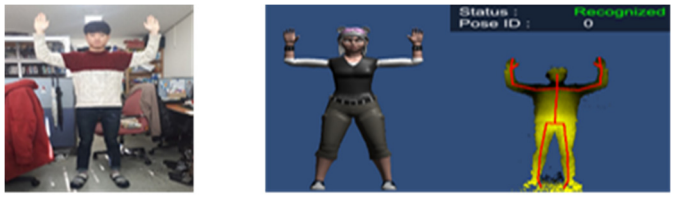
Person ID Participated in		Recognition result (%)						
		Pose ID						
Learning	Testing	0	1	2	3	4	5	6
BCDEFGH	A	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ACDEFGH	B	**57.30**	100.00	100.00	100.00	100.00	100.00	100.00
ABDEFGH	C	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ABCEFGH	D	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ABCDFGH	E	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ABCDEGH	F	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ABCDEFH	G	98.40%	100.00	100.00	100.00	100.00	100.00	100.00
ABCDEFG	H	100.00	**56.99**	100.00	100.00	100.00	100.00	100.00

Table 4. The results for the data include the ambiguous poses (person H)

Inputed pose ID	Recognized pose ID						
	0	1	2	3	4	5	6
0	100						
1	43.01	56.99					
2			100				
3				100			
4					100		
5						100	
6							100

4.2 The Results and Performance Analysis

Table 3 depicts the results of the test. It shows perfect accuracy in recognition for the most of test data. It is evidence that the proposed method can correctly discriminate the posture even though the motion data instantaneously includes instable joint information and outlier. Sensitivity of k value is examined by changing the k value from 1 to 50. The results say that the proposed method is not sensitive to the parameter k. The proposed method demonstrated stable performance across a wide range of k values, specifically showing consistent accuracy between k =5 and k =20. This indicates that the feature space is well-clustered and robust against parameter tuning. The figure 6 shows that poses are well classified and correctly recognize the posture even though a part of estimated skeletal data is unstable and incorrect (shown in figure 6-(b)). Furthermore, to check the accuracy of the system's perception of the user's unclear pose, we conducted another test, namely the intended pose error injection test. Pose data B and H with indexes contain ambiguous poses for Pose 0 and Pose 1. Table 4 clearly states that it is caused by input motion data containing ambiguous actions. It shows high recognition rates except for Pose 0 of B with human index and Pose 1 of H with human index. The only reason for misleading in certain poses is the similarity between the subject's pose 0 and pose 1. Most important consideration in posture recognition, not motion recognition, is transition motion from one posture to another. For examples, when the user changes the posture from a pose 3 to the pose 4, as shown in the figure 7, recognition results are flatted between the operation 3 and the operation 4. To prevent these phenomena, the system postpones the decision until the correlation coefficient downs to a threshold (t) and the number of the class are exceeded 80% within the k-th nearest neighbor in the middle of pose transition. Finally, we measured the computational efficiency to ensure applicability in real-time VR. The system was tested on a general PC. Frame rate is approximately 60 fps and latency time is 50 msec. from capture to classification.



(a) Example of stable posture

(b) Example of unstable posture

Figure 6. Real-time test : Most posed are well classified even if the posture is unstable and incorrect

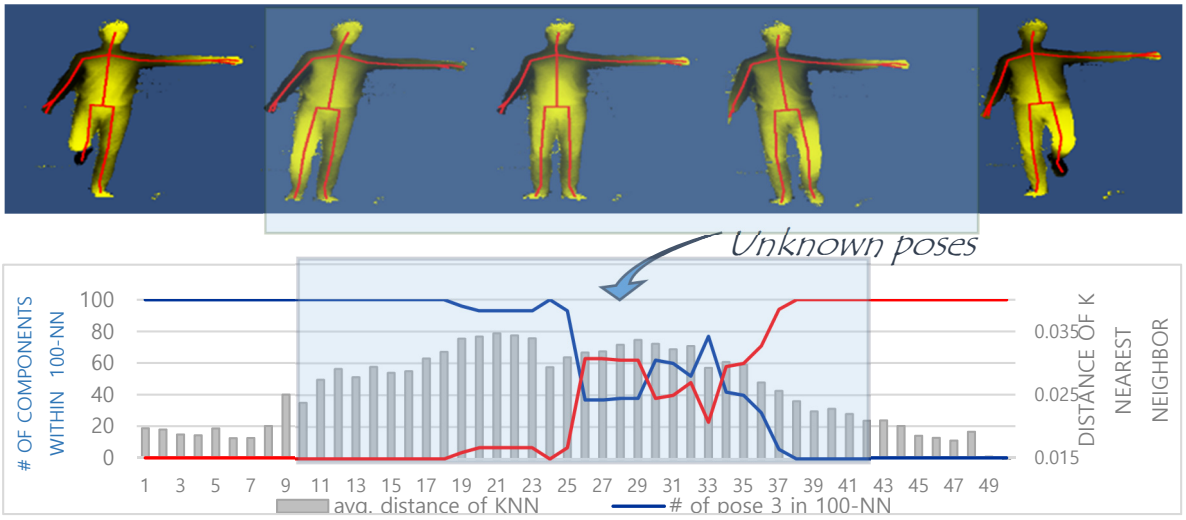


Figure 7. Transformation of components ratio during the motion capture

5. Conclusions

This paper presents a feature-engineering-based posture recognition framework tailored for non-wearable VR interfaces, specifically focusing on VR Skydiving applications. Unlike the prevailing trend of utilizing computationally intensive deep learning models, our method proposes a combination of Correlation-based Feature Vectors and Displacement Vectors (derived from T-pose). This approach ensures robust recognition even with low-resolution depth sensors and effectively neutralizes individual variability in body proportions, such as height and limb length. The experimental results confirm that this research provides significant contributions in three key aspects. First, the proposed framework ensures high recognition robustness against the inherent limitations of low-cost sensors. Traditional non-wearable sensors often suffer from data instabilities, such as "jitter" or "partial occlusion," depending on lighting and environmental conditions. To address this, we introduced a specialized preprocessing stage that normalizes skeletal data based on a user-specific T-pose. This normalization effectively neutralizes variability caused by individual body proportions and sensor estimation errors, enabling stable and consistent posture recognition across diverse users and environments. Second, the system significantly enhances classification stability during motion transitions. A persistent challenge in posture recognition is the "flickering" phenomenon, where classification results fluctuate rapidly as a user moves from one pose to another. We mitigated this issue by implementing a "Postponement Strategy" integrated with a confidence-based filtering mechanism. By retaining the previous stable state until the correlation coefficient and the components ratio exceed predefined certainty thresholds, the system provides a seamless and reliable user experience, eliminating the visual and functional instability typically found in raw transition data. Third, the proposed method achieves exceptional computational efficiency, making it highly suitable for real-time applications without requiring high-end hardware. Unlike recent trends that rely on resource-intensive deep learning models and high-performance GPUs, our approach optimizes a lazy learning algorithm based on refined feature vectors. Experimental results on a standard consumer PC demonstrated a consistent frame rate of 60 FPS and an end-to-end latency of 50ms. These metrics confirm that our system can deliver high-performance posture interfacing even on standalone VR devices or low-spec systems, offering a practical and scalable solution for real-time interaction.

While the current dataset is focused on seven standard maneuvers specialized for indoor skydiving, future research will aim to expand the dataset to include a wider variety of application-specific behaviors. We also plan to conduct large-scale testing with a more diverse group of participants to verify the system's generalizability. Furthermore, we intend to investigate the integration of our lightweight representation with Graph Convolutional Networks (GCN) or Open Set Recognition technologies. Such advancements would allow the system to process "unknown" or "undefined" behaviors in real-time, significantly broadening the scope and applicability of the proposed framework in various immersive interaction domains.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two stream adaptive graph convolutional networks for skeleton based action recognition," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12026-12035, 2019.

[2] S. Yan, Y. "Xiong, and D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition," In AAAI, vol. 32, pp. 7444-7452, 2018.

[3] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," In proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1112-1121, 2020.

[4] W. Xin, R. Liu, Y. Liu, Y. Chen, W. Yu, and Q. Miao, "Transformer for Skeleton-based action recognition: A review of recent advances," Neurocomputing, vol. 537, pp. 164-186, 2023.

[5] E. Cippitelli, S. Gasparrini, E. Gambi, and Susanna Spinsante, "A Human Activity Recognition System Using Skeleton Data from RGBD Sonsors," Hindawi Publishing, vol. 2016, Article ID: 4351435, 2016.

[6] J. Luo, W. Wang, and H. Qi, "Spatio-temporal feature extraction and representation for RGB-D human action recognition," Pattern Recognition Letters, vol. 50, pp.139-148, 2014.

[7] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "Space-time pose representation for 3D human action recognition," New Trends in Image Analysis and Processing, vol. 8158 of Lecture Notes in Computer Science, pp. 456-464, Springer, Berlin, 2013.

[8] S. Gaglio, G. Lo Re, and M. Morana, "Human activity recognition processing using 3-D posture data," IEEE Transactions on Human-Machine Systems, vol. 45, no. 5, pp. 586-597, 2015.

[9] V. Veeriah, N. Zhuang, and G. J. Qi, "Differential Recurrent Neural Networks for Action Recognition," Proceedings of IEEE Int'l Conf. on Computer Vision, pp. 4041-4049, 2015.