Current Science
CS 5 (4), 3408-3419 (2025)
Received 08 April 2025 | **Revised 11 May** 2025 | **Accepted 13 June**. 2025 | **Online Available 02 July** 2025
https://doi.org/ 10.52845/CS/2025-5-4-7

Current Science

Current Science

OPEN ACCESS

ISSN(O) 2795-8639

Original Article

DAF Filter: Double Adaptive Factors Filter for Dithering the Jitter in Virtual Model Interaction

Dayu Wu[1], Junjian Huang[1*]

[1]College of Electronic and Information Engineering, Southwest University, Chong Qing, China

[*]Corresponding Author: Junjian Huang

Abstract:
Nowadays, gesture recognition has been sufficiently developed and used in Human-Computer Interaction, the traditional way is to use the wearable device such as digital gloves. With the devel- opment of computer vision, control and interaction by gesture can provide the users with a more intuitive understanding and make it easier for the operator to present. However, most methods are based on specific device such as depth sensor, which contributes to gesture recognition cannot be extensively implemented, using a normal RGB camera to track the hand and recognize the gesture becomes a significant issue. Model-driven methods is the most low-cost way. However, a problem with this method is the occurrence of jitter during the model matching process. This paper proposes a improved filter to eliminate jitter while gesture recognition and hand tracking for virtual model interaction based on a monocular camera.

Keywords: Virtual reality, Gesture recognition, Hand tracking, Virtual model interaction

1. Introduction

As computers become increasingly prevalent in society, the develop- ment of Human-Computer Interaction (HCI) technology is positively influencing their use. As a branch of interaction design, major VR head- set manufacturers have unanimously adopted virtual reality controllers as the standard mode of interaction. These controllers are character- ized by independent operation of both hands. However, the use of controllers has several limitations, such as high economic costs, higher learning costs, and inconvenience in usage.

The ultimate goal is to make HCI as natural as interacting with other humans, with the development of artificial intelligence and computer vision, some different ways that utilize body posture information have occurred, such as facial recognition R et al. (2022), gesture recognition, eye tracking Peterson et al. (2020), and posture recognition. What is to emphasize, hand gestures are one of the most intuitive and com- mon forms of communication Tsai et al. (2020), for a long time, gestures have been considered a form of interaction technology that can pro- vide a more natural, creative, and intuitive way to communicate with computers. Gestures are a form of non-verbal communication, typically defined as various postures and movements of the hands or hands in combination with the arms, used to convey ideas, emotions, or emphasis Goldin-Meadow (1999). Gestures are an interactive means that aligns with human daily habits. In everyday life, people often use gestures to communicate information or express specific intention. Therefore, integrating gestures into HCI is an important research area.

Gesture recognition based on touch sensors typically relies on data gloves equipped with multiple sensors. In 1983, Grimes pioneered the invention of the earliest data gloves Grimes (????). In 2004, Kevin and his team designed a wireless instrument glove called CyberGlove II for gesture recognition Kevin et al. (2004). In 2022, Flexible Strain Sensors for Wearable Hand Gesture Recognition had been presented, which of- fers the possibility of directly measuring the finger motion behaviors for accurate and cost-effective hand gesture recognition Si et al. (2022). Md. Ahasan Atick Faisal came out with a cost-effective dataglove in- cludes five flex sensors, an inertial measurement unit, and a powerful microcontroller that handles onboard processing and facilitates wireless connectivity which can recognize static gesture with 82.19% accuracy and dynamic gesture with 97.35% accuracy by using their parallel-path neural network architecture for handling multimodal data Faisal et al. (2022).

But it is still inconvenient and uncomfortable to use the gloves to

detect the gesture, we prefer to tracking the hand without any wear- able device. Firstly , an approach which use colored gloves to make the computer detect specific region of hand from a monocolor picture came out in 1996 Iwai et al. (1996). With recent advancement of com- puter vision, real-time solutions of detect gesture under the option of colored gloves were presented Wang and Popović (2009) Lamberti and Camastra (2011). After that, computer vision-based gesture recognition methods have seen an explosion in development. Non-direct contact gestures become a new available way to interact with virtual world LI et al. (2019). Generally speaking, there are seven kinds of approaches are widely used, they are skin color recognition, appearance recognition, motion recognition, skeleton recognition, depth recognition, 3D model recognition and deep learn recognition Oudah et al. (2020).These meth- ods are in correspond with different types of cameras, such as RGB camera, time of flight (TOF) camera Kudic and Jokic (2023) , thermal cameras Breland et al. (2021), and different kinds of algorithms, it can be divided into two genres, data-driven algorithms and model-driven algorithms.

However, although the technology has become relatively mature,

there have been some issues on both the hardware and software sides that have affected the popularization of gesture recognition in human-computer interaction. In terms of hardware, for example, the Eva Kollorz's experiment result shows the overall classification rate is 93.14% without and 94.61% with depth features Kollorz et al. (2008), in other words, the difference in effect is not significant, but the cost of TOF camera and webcam differs a lot. So single RGB camera is adapted to this system in this paper.

In terms of algorithm, algorithms can be broadly categorized into two main types: data-driven algorithms and model-driven algorithms.

Data-driven gesture recognition algorithms refer to the utilization of collected data, where training samples are paired with their correspond- ing labels, enabling machine learning to map from samples to labels. Such algorithms fall under the category of discriminative approaches. The advantages of these methods lie in their avoidance of complex model design, while their drawback lies in the requirement for large amounts of data. However, in the current era of big data, the quantity of data is no longer an issue, and this data-driven approach has become the prevailing research direction.

Model-driven gesture recognition algorithms are a more precise ap- proach. The fundamental idea behind these algorithms is to match object models with images, thereby inferring the object's position and orientation. The advantages of this algorithm lie in its high accuracy. However, its drawback is the requirement to pre-establish object mod- els. Neural network-based gesture recognition algorithms, which are emerging as a novel development direction, are based on model-driven gesture recognition algorithms. Real-Time Hand Gesture Detection and Classification in video using Convolutional Neural Networks was pre- sented in 2017 Ge et al. (2018), estimating 3D gesture poses from a single RGB image was presented in 2019 Ge et al. (2019), real-time tracking of gesture models using a single RGB camera in 2020, the re- duction in device requirements has made low-cost real-time gesture tracking feasible.

Reducing the cost of gesture recognition is a subject that warrants

scholarly investigation. Data-driven algorithms that require handling substantial amounts of data evidently lack a price advantage. Conse- quently, despite data-driven gesture recognition algorithms being the mainstream direction of development, model-driven gesture recogni- tion algorithm is still popular. Gesture recognition based on RGB images becomes a workable solution to save cost.In 2020, Google presented a real-time on-device hand tracking solution that predicts a hand skeleton of a human from a single RGB camera for augmented reality(AR)/virtual reality(VR) applications.

Gesture Recognition is one part of object detection. There are other detect algorithms, such as Openpose Marknez (2019), You Only Look Once (YOLO) Terven and Cordova-Esparza (2023), compared with Openpose and YOLO, mediapipe targets a single person, and it runs on a mobile CPU Ienaga et al. (2022), so it is suitable for virtual model interaction.

Mideapipe Hands Zhang et al. (2020) developed by GOOGLE utilizes a hand detection model (BlazePalm) to detect the palm region, a hand landmark model to accurately locate hand keypoints, and a gesture clas- sifier for gesture classification. This approach allows for lightweight and efficient hand gesture recognition using a single RGB camera.

The main contributions of this paper are as follows:

1. Creating a virtual interaction system based on RGB gesture recog- nition. The system consists of two components. The gesture recog- nition and hand tracking part are developed using the Python pro- gramming language along with OpenCV and Mediapipe Hands. The other component is based on Unreal Engine. These two components communicate with each other by exchanging information through the User Datagram Protocol (UDP).
2. Introducing the OneEuro Filter to address the jitter issue during hand movement. The process of hand movement is sampled, and the sampled results are processed to achieve a smoothing effect through the processing of the sampled sequence. This ensures good stability in gesture control under both fast movement and slow adjustment scenarios.
3. Presenting DAF filter, which is based on OneEuro filter algorithm. DAF filter can improve the performance under the situation where dithering the jitter while hand tracking, the test result shows the modification performs better than OneEuro Filter.

2. Related Work

Natural jitter during hand movement lies in the presence of small, direc- tionally random, high-frequency movements between frames. The first step is to detect the motion direction between image frames. Actually, this step is similar to video stabilization task, so this section intro- duced video stabilization work. Currently, the commonly used solutions include the following:

Corner detection: Any object in an image usually contains unique fea- tures but is often composed of a large number of pixels. Corner points are a smaller set of points that can accurately describe the object. Cor- ner detection algorithms can analyze and identify the most prominent feature points in an image, which are used for object recognition and tracking. Common corner detection algorithms include: Harris Corner Detector, Shi-Tomasi Corner Detector, Features from Accelerated Seg- ment Test(FAST) Corner Detector. Some recent advancements in corner detection are as followed: IPCSWan et al. (2023) use intensity, pattern, curvature, and scale dimensions to ensure the performance of the cor- ner detector. RCDSCZhang et al. (2022) is a novel method for corner detection based on the ratio of center distances of symmetric contour by discrete curvature estimation.

Optical flow: The movement of image objects between two consecu- tive frames caused by the movement of the target object is called optical flow. It is a 2D vector field that can be used to represent the movement of a point from the first frame to the second frame. As a commonly used method for motion displacement monitoring in various fields, opti- cal flow algorithms have multiple implementations and are continuously being improved. Such as Lucas-Kanade optical flow algorithm and its improved versionAl-Qudah and Yang (2023), Liu et al. (2023), Zhong et al. (2023), Horn-Schunck optical flow algorithm and its improved ver- sionElasri et al. (2024), Johnson et al. (2022). The Lucas-Kanade optical flow algorithm estimates optical flow vectors by minimizing error within a local window of the image, making it suitable for detecting small-scale motion.

Random Sample Consensus (RANSAC): The RANSAC algorithm is rel- atively simple, easy to understand, and implement. It consists of several basic steps: random sampling, model fitting, error calculation, and model selection. RANSAC can iteratively estimate the parameters of a math- ematical model from a set of observational data that includes outliers. For two consecutive frames of images, each having its own set of cor- ner points, RANSAC can discover matching point sets from noisy data, thereby calculating the transformation matrix for the two frames of images. Based on RANSAC, other improved algorithms such as PSC- RANSACMa et al. (2021), RANSAC method combined with the total least squares methodWu et al. (2022), Crossline Correction Non-linear RANSAC (CCNL-RANSAC)Yang et al. (2021) which has shown good performance in detecting curved objects

3. Use Filter to Dither the Jitter

Among all the human-computer interaction devices currently available, the mouse is undoubtedly the most convenient and efficient device. In the gesture interaction approach used in this paper, the part that con- trols the rotation of the virtual model is actually achieved by using hand tracking to control the cursor Guliani et al. (2023), known as gesture mouse. Various body parts can be tracked to manipulate the cursor, such as the wrist, arm, and head. However, finger gestures have been found to possess the highest sensitivity, accuracy, and convenience. Using fin- ger gestures to control the cursor achieves a accuracy that is 3.8 times Card et al. (1991) higher and a speed that is four times faster than mouse control.Nevertheless, there are challenges in the process of gesture- controlled cursor movement. During the tracking of hand gestures, the speed of hand movement can vary from fast to slow. This requires rapid adjustments and fine-tuning of the displayed model's position. As a re- sult, the frequency bandwidth of the gesture mouse output signal is wide. In situations that require quick movements, the filter must out- put signals with low latency to provide users with a better experience. On the other hand, in situations that require slow movements, the filter must be able to eliminate high-frequency jitter from the signal Casiez et al. (2012).

3.1 Methods Comparison and Selection

In statistics, moving average is a computational method used to analyze data points by creating a series of average values from different subsets of the entire dataset. It is a commonly used tool in technical analysis for analyzing time series data. Moving averages are typically used in con- junction with time series data to smooth out short-term fluctuations and highlight long-term trends or cycles. The threshold between short- term and long-term depends on the application, and the parameters of the moving average are set accordingly.

Exponential smoothing LaViola (2003) is a commonly used method in production forecasting. The commonly used exponential smoothing methods include single exponential smoothing and double exponential smoothing.

Single exponential smoothing, also known as simple exponential smoothing, refers to the application of a single level of smoothing. Sin- gle exponential smoothing combines the forecasted values for a given period with the observed values through a linear combination to obtain the forecasted value for period $t + 1$.

The definition of the filter is as followed:

$$s_i = \alpha \cdot x_i + (1 - \alpha) \cdot s_{i-1} \quad (1)$$

$$x_{i+h} = s_i \quad (2)$$

Double exponential smoothing is a method that applies exponen- tial smoothing twice to the values obtained from single exponential smoothing. It cannot be used for forecasting independently; it must be combined with single exponential smoothing to establish a mathemat- ical model for forecasting and then utilize the mathematical model to determine the forecasted values.

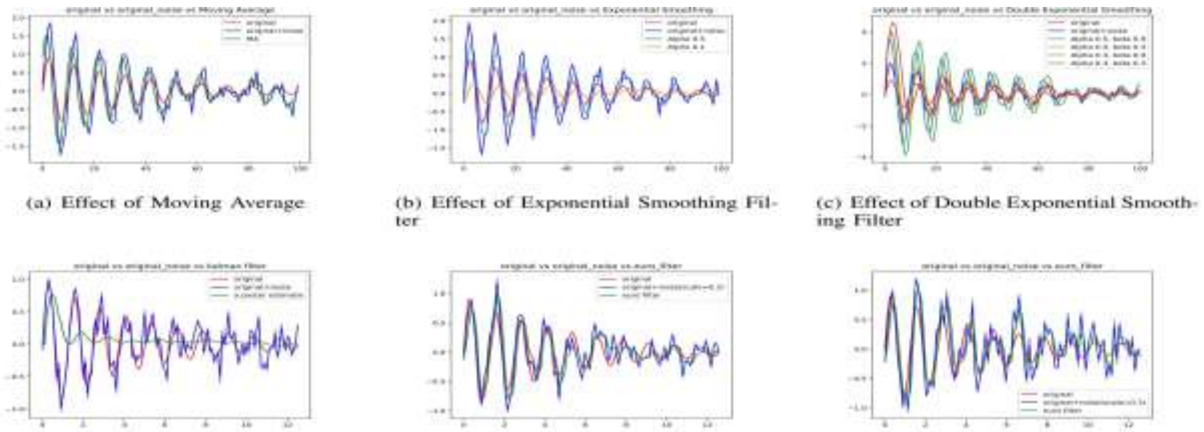
The definition of the filter is as followed:

$$s_i = \alpha \cdot x_i + (1 - \alpha) \cdot (s_{i-1} + t_{i-1}) \quad (3)$$

$$t_i = \beta \cdot (s_i - s_{i-1}) + (1 - \beta) \cdot t_{i-1} \quad (4)$$

$$x_{i+h} = s_i + h \cdot t_i \quad (5)$$

Kalman filtering Welch et al. (1995) is an algorithm that utilizes the linear system state equations and input-output observation data to achieve optimal estimation of the system state. From the simulation re- sults, we can observe that the performance of the Kalman filter is heavily influenced by the previous states. Throughout the entire simulation pro- cess, the performance of the Kalman filter is not satisfactory. Although it exhibits good noise removal capabilities, it has a significant impact on the amplitude of the input signal. The Kalman filter is unable to provide accurate predictions for the nonlinear motion trajectories that are being addressed in this paper.



(a) Effect of Moving Average

(b) Effect of Exponential Smoothing Filter

(c) Effect of Double Exponential Smoothing Filter

(d) Effect of Kalman Filter

(e) Effect of OneEuro Filter(noise scale=0.2)

(f) Effect of OneEuro Filter(noise scale=0.3)

Figure 1 Comparison of Effect of Different Filters

The inspiration of OneEuro filter comes from RC low-pass filtering circuit. When the fewer high frequency signal is needed,

The definition of the filter is as followed:

$$\hat{X}_1 = X_1 \qquad (6)$$

$$\hat{X}_i = \alpha X_i + (1-\alpha)\hat{X}_{i-1}, \quad i \geq 2 \qquad (7)$$

Filters utilize the exponential function for smoothing, the smoothing factor is not a constant but self-adaptive, calculated dynamically using information about the rate of signal variation. The adaptive smoothing factor aims to strike a balance between jitter and lag, as humans are sensitive to jitter at low speeds and more sensitive to lag at high speeds.

The definition of smoothing factor is as followed:

$$\alpha = \frac{1}{1+\frac{\tau}{T_e}} \qquad (8)$$

$$T_e = T_i - T_{i-1} \qquad (9)$$

$$\tau = \frac{1}{2\pi f_c} \qquad (10)$$

$$\hat{X}_i = \left(X_i + \frac{\tau}{T_e}\hat{X}_{i-1}\right)\frac{1}{1+\frac{\tau}{T_e}} \qquad (11)$$

$$f_c = f_{c_{min}} + \beta \, \dot{\hat{X}}_i \qquad (12)$$

Because Te can be computed automatically, so the only configurable parameter is fc , to make fc correspond with input signal, fc should be in- fluenced by the rate of signal variation, Based on reality, the initiation and cessation of gesture movements can occur at varying speeds, neces- sitating changes in the rate of variation to proceed relatively smoothly in all cases. Therefore, the rate at a particular moment must change based on the previous moment's rate, ensuring overall continuity. Equation shows how $\hat{X}_i$ changes.

The decaying oscillation curve possesses both fast high-frequency oscillation and slow low-frequency oscillation characteristics, which can be used to simulate rapid and large-scale hand movements as well as slow and small-scale adjustments. Therefore, the decaying oscilla- tion signal is chosen as the simulation of hand motion to test the performance of various filters.

The runtime of all filters are as followed, the simulation environment consists of the 13th Gen Intel(R) Core(TM) i5-13600KF processor and Python 3.9.

The definition of rate is as followed:

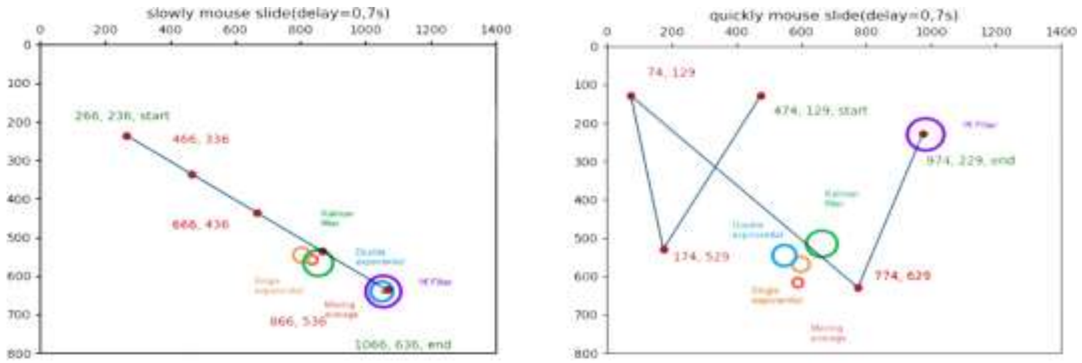
$$Rate = \frac{T_{currentfilter}}{T_{OneEurofilter}} \tag{13}$$

Table 1 Respone Time of Filters

Name	Time(ms)	Rate
Moving Average	0.2674	171.53%
Single Exponential Smoothing	0.3176	203.86%
Double Exponential Smoothing	0.3520	225.93%
Kalman Filtering	0.3379	216.88%
OneEuro Filtering	0.1558	100%

Analyzing the simulation results, Exponential smoothing, Double Ex-ponential smoothing, and Kalman filtering are capable of obtaining well-defined waveforms, referred to as the first category of waveforms. On the other hand, Moving average and OneEuro Filtering are able to gen- erate waveforms that are roughly similar to the input waveform, known as the second category of waveforms. While the first category wave- forms exhibit good quality, they require significantly more processing time compared to the second category waveforms. The second cate- gory waveforms closely match the input waveform and provide faster response times. They are particularly effective in accurately matching waveforms in low-amplitude sections, satisfying the requirements of gesture control. OneEuro Filtering, in particular, excels in terms of response time.

Mouse movement controlled by program was processed by different filters, as Fig. 2 shows, the speed different OneEuro filter behaves well in both quick mouse movement and slow mouse movement.



(a) Different filters effect when the mouse slides slowly

(b) Different filters effect when the mouse slides quickly

Figure 2 Comparison of Different Movement States

3.2 Method Improvement

However, the result of OneEuro Filter can still be improved, as men- tioned above, there are three factors influence the result of filtering, OneEuro Filter only transform α into adaptive factor, to gain better re- sult, β and minimum cutoff frenquency are also adjusted into adaptive factors in this system.

β factor which is the coefficient of the speed, as the definition of OneEuro Filter and the situation the system faced, when the speed of movement is high, the movement is less infected by noise, under this circumstance, the speed dominates the waveform, while the speed of movement is low, the movement is greatly infected bu noise so β should be used to control the effect of speed, as the analysis above, Range is selected to be β factor. The range of data in specific time period varies with the speed.

The definition of rate is as followed:

$$\beta i = Ximax - Ximin \qquad (14)$$

To value the effect of improvement, Mean Squared Error (MSE) which is a common method to measure the fit between two curves is introduced. By computing the square of the differences between cor- responding points on the two curves, smaller MSE indicates a better fit.

The definition is as followed:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_{test}^{(i)} - \hat{y}_{test}^{(i)})^2 \qquad (15)$$

A test is arranged to proof the effect. In the test, MSE of the original OneEuro Filter which uses fixed β factor and original signal is called original MSE while the improved one with adaptive β factor is called improved MSE. The definition of improved rate is as followed:

$$Improved\ Rate = \frac{Original\ MSE}{Improved\ MSE} - 1 \qquad (16)$$

The result of test is as TABLE II shows:

Table 2 The comparison of different method's method with MSE

Original MSE	Improved MSE	Improved Rate
0.0831	0.0297	179.80%
0.0859	0.0300	186.33%
0.0833	0.0361	130.75%
0.0863	0.0437	97.48%
0.0866	0.0347	149.57%
0.0822	0.0317	159.31%
0.0914	0.0365	150.41%
0.0799	0.0397	101.26%
0.0987	0.0412	139.56%
0.0896	0.0499	79.56%
0.0805	0.0227	254.63%
0.0874	0.0410	113.17%
0.0819	0.0227	260.79%
0.0837	0.0330	153.64%
0.0792	0.0256	209.38%

Algorithm 1 Alpha computation

Input:
Data update rate in HZ: *rate*
Cutoff Frequency in HZ: *cutoff*
Output:
Alpha value for low-pass filter
$tau \longrightarrow 1 / (2 * \pi * cutoff)$ $te \longrightarrow 1 / rate$
return $1 / 1 + (tau / te)$

Algorithm 2 Beta computation

EXT:
Sampling sequence: *Seq*
Length of Sampling sequence: *h*
Input:
Mouse movement value: *x*
Adjust factor: *a*
Output:
Beta value for low-pass filter
for $i = 0$ to h **do**
if *Seq* is not full **then**
$Seq \longrightarrow x$
else
Clear *Seq*
end if end for
$Range = Seq_{max} - Seq_{min}$
return $Range * a$

4. Function Implementation

In this section, OneEuro Filter is implemented into the visual model interaction. In order to keep a balance in speed and accuracy, an experi- ment is designed to select sampling frequency. In response to different scenarios, different cases are designed to address them.

4.1 Sampling Sequence

The data input to the DAF Filter should be a sequence, gesture recogni- tion is a real-time process. Therefore, we use a window of data over a specific time interval and take the average value as the coordinate for a particular point in time. If the sampling sequence within this time win- dow is too long, it can result in slower data processing and noticeable delays for the user.

After test, when the sequence length longer than 32, the process re- sult is stable and satisfied. Under the hardware mentioned above, the test result indicates when the length is 128, the average time is ms, which can meet the requirement of 30 frames.

The test result is as Table 3 shows:

Table 3 Test result

	Length	Average Time(ms)
	32	0.497
	128	1.986
	256	3.973

Based on the experiments conducted, it was observed that when using 4 sampling points, the mouse movement achieves a relatively smooth effect. According to the experimental results, each iteration of the algorithm takes approximately 0.5ms to execute. If the sampling points are increased to 8, the time required becomes 1ms, and the mouse movement experiences a noticeable lag. Therefore, the decision was made to stick with the 4-point sampling method as the final processing approach.

4.2 Case in Different Situation

There will be these cases occur in use:

CASE 1: Only one hand detected

In this case, no matter which hand can only operate to change the model location, if the camera detects the second hand and the two hands are a combination of left hand and right hand, the program will switch to case2 automatically.

CASE 2: A combination of left hand and right hand detected

In this case, left hand will always be the controller of material, right hand will always be the controller of model rotation.

CASE 3: Two same hands or more than two hands detected

Program cannot be controlled by gesture until the hands detected meet the confidence of case1 or case2.

Because the screen is 2D, so there are only 2 degrees of freedom can be expressed. Take the reality display acquirement into account, we choose yaw and pitch to adjust. When it comes to how to transmit the location coordinate into the rotation coordinate, we choose the tip point of the right index finger whose index is 4 in landmark array as the reference point, then we calculate the rotation angle according to the location coordinate of the reference point.

The index controls which material will be applied to the mesh body. We choose the gesture of left hand to control the material switch. When only one hand detected, the first material will be applied to the mesh body as default.

After the hands detected, the hand landmark will be corresponded to the model and the coordinate will be estimated, rotation coordinate will be sent to Unreal Engine according to UDP transmission. UDP transmis- sion can only transmit text, so we set the coordinate format as “[yaw, pitch, index]” so Unreal Engine can recognize the string and transform the character format data into integer or floating point number and set the rotation angle of the mesh body.

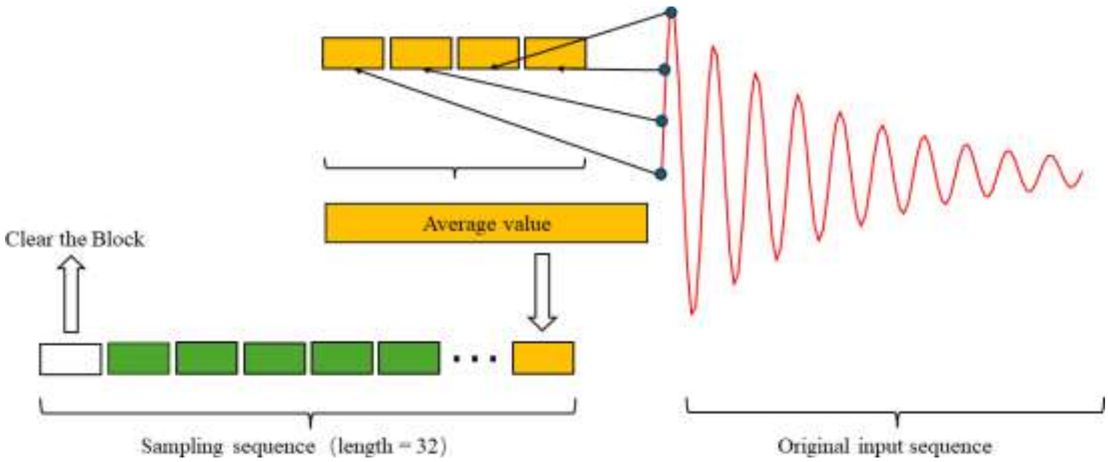


Figure 3 Sampling Sequence

5. Conclusion and Future Scope

The system’s achievements are demonstrated through various ways:

Virtual Model Interaction: The system enables users to interact with virtual models using hand gestures captured by a camera. Users can manipulate and control the virtual models in real-time, providing an immersive and interactive experience, virtual model contains various types.

Real-time Feedback and Stability: The system employs techniques like the OneEuro filter for filtering and stabilizing the hand movement data. This ensures that the interaction is smooth, stable, and responsive, providing users with a seamless experience.

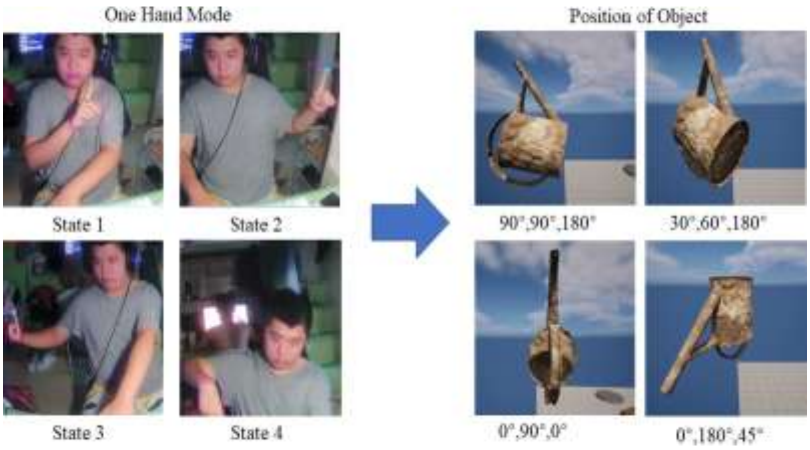


Figure 4 Using One Hand to Control Position of Object Under One Hand Mode

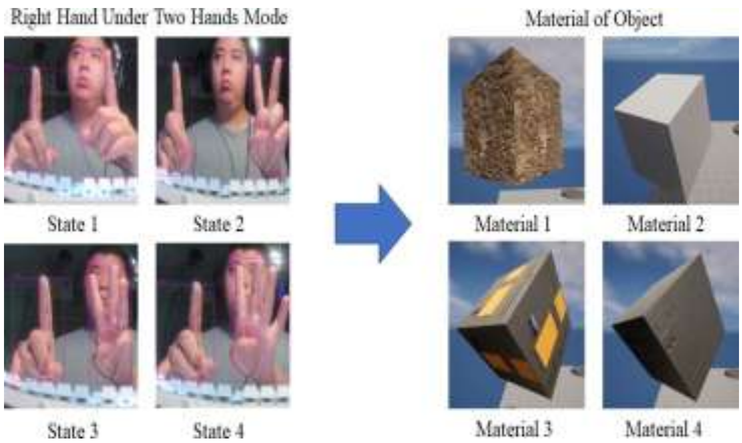


Figure 5 Using Right Hand to Control Material of Object Under Two Hands Mode

Low-Cost and Accessibility: The system offers a cost-effective solu- tion for virtual interaction, as it utilizes commonly available hardware components and open-source libraries. This makes it accessible to a broader range of users and facilitates its widespread adoption in vari- ous domains, such as industrial product customization and educational applications.

Overall, the system's achievements are showcased through its ability to provide a user-friendly, immersive, and cost-effective virtual inter- action experience, enabling users to engage with virtual models in a natural and intuitive way.

In various fields, this virtual display approach is an excellent method for providing users with an intuitive experience. As mentioned earlier, it can be applied in customizing industrial products, as well as in education where students can gain a visual understanding of abstract concepts through such methods. The low-cost aspect of this approach makes it more accessible and widespread.

Looking ahead, there are several potential future directions and advancements for the system:

Enhanced Gesture Recognition: Continued research and develop- ment can focus on improving the accuracy and robustness of gesture recognition algorithms. This could involve incorporating machine learn- ing techniques to train models on larger and more diverse datasets, resulting in more precise and reliable gesture recognition.

Multi-user Collaboration: Enabling multi-user collaboration within the virtual environment would allow multiple users to interact with the same virtual models simultaneously. This could facilitate collaborative design and planning tasks, training simulations, and virtual meetings, providing a more immersive and collaborative experience.

Cross-Platform Compatibility: Ensuring compatibility with different platforms and devices, including mobile devices and various operating systems, would expand the system's reach and accessibility. This would enable users to engage with virtual models and environments using their preferred devices, enhancing convenience and user adoption.

Overall, the future of the system lies in advancing gesture recog- nition, expanding functionality, incorporating new technologies, and improving user experience, ultimately enabling more immersive and intuitive virtual interactions in various domains.

References

1. Al-Qudah, S. & Yang, M. (2023) Large displacement detection using improved lucas–kanade optical flow. *Sensors*, 23(6), 3152.
2. Breland, D.S., Dayal, A., Jha, A., Yalavarthy, P.K., Pandey, O.J. & Cenkeramaddi, L.R. (2021) Robust hand gestures recognition us- ing a deep cnn and thermal images. *IEEE Sensors Journal*, 21(23), 26602–26614. doi:10.1109/JSEN.2021.3119977.
3. Card, S.K., Mackinlay, J.D. & Robertson, G.G. (1991) A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems (TOIS)*, 9(2), 99–122.
4. Casiez, G., Roussel, N. & Vogel, D. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: *Proceed- ings of the SIGCHI Conference on Human Factors in Computing Systems*, *2012*, pp. 2527–2530.
5. Elasri, A., Cherroun, L. & Nadour, M. (2024) Robotic visual-based navigation structures using lucas-kanade and horn-schunck algo- rithms of optical flow. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*,, 1–24.
6. Faisal, M.A.A., Abir, F.F., Ahmed, M.U. & Ahad, M.A.R. (2022) Exploit- ing domain transformation and deep learning for hand gesture recognition using a low-cost dataglove. *Scientific Reports*, 12(1), 21446.
7. Ge, L., Liang, H., Yuan, J. & Thalmann, D. (2018) Robust 3d hand pose estimation from single depth images using multi-view cnns. *IEEE Transactions on Image Processing*, 27(9), 4422–4436.
8. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J. et al. 3d hand shape and pose estimation from a single rgb image. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019*, pp. 10833–10842.
9. Goldin-Meadow, S. (1999) The role of gesture in communica- tion and thinking. *Trends in Cognitive Sciences*, 3(11), 419–429. doi:https://doi.org/10.1016/S1364-6613(99)01397-2.
10. Grimes, G.J.) *Digital data entry glove interface device*.
11. Guliani, S., Nagpal, R., Sehgal, R. & Singhal, A. Gesture controlled mouse navigation: Hand landmark approach. In: *2023 13th Inter- national Conference on Cloud Computing, Data Science & Engineer- ing (Confluence). IEEE*, *2023*, pp. 183–188.
12. Ienaga, N., Takahata, S., Terayama, K., Enomoto, D., Ishihara, H., Noda, H. et al. (2022) Development and verification of postural control assessment using deep-learning-based pose estimators: Towards clinical applications. *Occupational Therapy International*, 2022.
13. Iwai, Y., Watanabe, K., Yagi, Y. & Yachida, M. Gesture recogni- tion using colored gloves. In: *International Conference on Pattern Recognition*, *1996*.
14. Johnson, B., Thomas, S. & Rani, J.S. (2022) High throughput archi- tecture for multiscale variational optical flow. *Journal of Real-Time Image Processing*, 19(3), 675–686.
15. Kevin, N.Y.Y., Ranganath, S. & Ghosh, D. Trajectory modeling in gesture recognition using cybergloves and magnetic trackers. In: *Tencon IEEE Region 10 Conference, 2004*.
16. Kollorz, E., Penne, J., Hornegger, J. & Barke, A. (2008) Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4), 334–343. Kudic, H. & Jokic, D. (2023) Highly reliable time-of-flight l-shaped gesture recognition system. *European Journal of Engineering and Technology Research*, 8(2), 69–72.
17. Lamberti, L. & Camastra, F. Real-time hand gesture recognition us- ing a color glove. In: *Image Analysis and Processing–ICIAP 2011: 16th International Conference, Ravenna, Italy, September 14-16, 2011, Proceedings, Part I 16. Springer*, *2011*, pp. 365–373.
18. LaViola, J.J. Double exponential smoothing: an alternative to kalman filter-based predictive

tracking. In: *Proceedings of the workshop on Virtual environments 2003,2003*, pp. 199–206.

19. LI, Y., HUANG, J., TIAN, F., WANG, H.A. & DAI, G.Z. (2019) Gesture interaction in virtual reality. *Virtual Reality Intelligent Hardware*, 1(1), 84–112.
20. Liu, X., Tang, J., Shen, C., Wang, C., Zhao, D., Guo, X. et al. (2023) Brain-like position measurement method based on improved optical flow algorithm. *ISA transactions*, 143, 221–230.
21. Ma, S., Guo, P., You, H., He, P., Li, G. & Li, H. (2021) An image match- ing optimization algorithm based on pixel shift clustering ransac. *Information Sciences*, 562, 452–474.
22. Marknez, G.H. (2019) Openpose: Whole-body pose estimation. Ph.D. thesis, Carnegie Mellon University Pittsburgh, PA, USA.
23. Oudah, M., Al-Naji, A. & Chahl, J. (2020) Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging*, 6(8). doi:10.3390/jimaging6080073. URL https://www.mdpi.com/2313-433X/6/8/73
24. Peterson, M., Tober, B., Littlejohn, D. & Hill, M. (2020) Anticipating gaze-based hci applications with the tech receptivity interval: Eye tracking as input. *Visible Language*, 54 (1/2), 99–127.
25. R, S., K, S., K, A.P., Vishwagna, A.B. & N, P. Mouse cusor control using facial movements - an hci application. In: *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022*, pp. 367–371.
26. Si, Y., Chen, S., Li, M., Li, S., Pei, Y. & Guo, X. (2022) Flexible strain sensors for wearable hand gesture recognition: from devices to systems. *Advanced Intelligent Systems*, 4(2), 2100046.
27. Terven, J. & Cordova-Esparza, D. (2023) A comprehensive re- view of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*,.
28. Tsai, T.H., Huang, C.C. & Zhang, K.L. (2020) Design of hand gesture recognition system for human-computer interaction. *Multimedia tools and applications*, 79, 5989–6007.
29. Wan, C., Cao, J., Wei, X., Huang, J., Chen, Z., Xu, D. et al. (2023) Ipcs: An improved corner detector with intensity, pattern, curvature, and scale. *The Visual Computer*, 39(7), 2499–2513.
30. Wang, R.Y. & Popović, J. (2009) Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, 28(3), 1–8.
31. Welch, G., Bishop, G. et al. (1995) An introduction to the kalman filter,.
32. Wu, Q., Liu, J., Gao, C., Wang, B., Shen, G. & Li, Z. (2022) Improved ransac point cloud spherical target detection and parameter estimation method based on principal curvature constraint. *Sensors*, 22(15), 5850.
33. Yang, K., Yu, L., Xia, M., Xu, T. & Li, W. (2021) Nonlinear ransac with crossline correction: An algorithm for vision-based curved cable detection system. *Optics and Lasers in Engineering*, 141, 106417.
34. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L. et al. (2020) Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*,.
35. Zhang, S., Liu, S., Zheng, Q. & Huang, M. (2022) An efficient cor- ner detector using ratio of center distance of symmetric con- tour technique. *Journal of Circuits, Systems and Computers*, 31(16), 2250286.
36. Zhong, L., Meng, L., Hou, W. & Huang, L. (2023) An improved vi- sual odometer based on lucas-kanade optical flow and orb feature. *IEEE Access*,.