Transactions on Computational and Scientific Methods | Vo. 6, No. 02, 2026
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Human-Machine Interaction Gesture Feature Recognition and Edge Repair Algorithm Research

Joshua Bennett
New York University, New York, USA
j.bnet92@outlook.com

Abstract: How to accurately detect and segment gesture parts from the facial area is a major research focus in the field of human-machine interaction. The similarity in texture between the hand and the face poses significant challenges to the stability of gesture recognition. This paper proposes a gesture recognition algorithm based on edge repair using a computer-based method to handle the issue of overlapping between hands and faces. Based on the Chamfer distance matching method, an edge repair algorithm is introduced to handle blurred edge areas at the intersec-tion of hands and faces. Experimental results show that the accuracy of the edge repair algorithm reaches 94.6%, significantly outperforming other methods with an accuracy of only 8.7%. This algorithm is particularly effective in scenarios involving frequent gesture changes and facial movement.

Keywords: Large language models; Agent; Prompt engineering; Chain-of-thought

1. Introduction

In the field of computer vision research, visual gesture recognition serves as a key discip-line within human-computer interaction. Its development is closely linked to the progress of human-computer interaction. In gesture recognition research based on computer vision technology, the accurate identification and segmentation of gestures in images or videos form the technical foundation. Image segmentation is a fundamental aspect of human-computer interaction studies and holds significant research value in computer visual ana-lysis.

Gesture segmentation is a fundamental issue in the field of gesture recognition. Many research directions exist in this area, where segmentation techniques rely on the advanced features of gestures such as fingertips, fingers, joints, and other specific parts. These features are used to determine the presence of a hand. Studies on human skin tone and texture have shown that the color and texture of the hand and face are very similar. The problem of facial and hand skin interference remains unsolved. At present, many studies focus on gesture localization or tracking, without providing precise segmentation of gesture regions [1][2].

In [3], a Haar-like feature classifier is used to detect and track hand gestures, suc-cessfully achieving stable tracking of the hand in front of the face. In [4], a model based on motion of facial target parts is established, combining skin color with local motion for dual tracking to detect crossing hands in front of the face. In [5], a Gaussian skin color model and density distribution features are used to track gestures for segmentation and recognition in specific regions. This method ensures high recognition accuracy for gestures entering and exiting open-face regions, enabling accurate distinction of hands and faces. In [6], gesture shapes, textures, and color features are extracted and Bayesian model-generated images are used for gesture segmentation. Although this method is highly adaptive to background complexity, it has slower computation speed.

Based on the foundation of previous research, this paper proposes a gesture recognition method using edge-repair-based segmentation of gesture parts. This method provides fast recognition, high accuracy, and has high research value for processing visual elements in human-computer interaction.

2. Gesture Recognition System Architecture

With the widespread deployment of modern network technologies and intelligent sensing devices, computer-based gesture recognition systems have gradually evolved toward modular, end-to-end architectures within human-computer interaction (HCI) frameworks. In designing such systems, emphasis is placed on robustness, scalability, and the integration of perception and decision-making modules.

Liu et al. [7] proposed a graph neural network-based framework for user satisfaction classification in HCI systems, demonstrating that structured feature representation and relational modeling can significantly enhance classification stability and system robustness. Inspired by this structured modeling paradigm, the architecture proposed in this study adopts a modular design to ensure reliable feature extraction and decision-making in dynamic gesture interaction scenarios.

The overall system architecture consists of four primary components: data acquisition, preprocessing, feature extraction and segmentation, and gesture classification. A camera captures real-time gesture information, and the acquired visual data are transmitted to the gesture recognition subsystem through the host computer. Within this subsystem, image preprocessing operations-including denoising, normalization, and edge enhancement-are first applied to suppress background interference and illumination variations.

Following preprocessing, the system performs automatic gesture region localization. Drawing on the structured feature modeling concept described by Liu et al. [7], the extracted visual features are organized into a digital modeling framework, enabling effective filtering and refinement of gesture-related information. This structured representation enhances the robustness of gesture classification under conditions of facial movement and hand-face overlap.

Subsequently, based on the predefined gesture models and the interaction interface requirements, the system classifies gesture information through feature matching and decision rules. The complete gesture recognition process, including acquisition, preprocessing, segmentation, feature modeling, and classification, is illustrated in Figure 1.

After the camera captures the input image, the system automatically identifies the gesture region and performs denoising preprocessing. Feature information is then recognized and filtered through digital modeling, forming the basis for subsequent edge-repair-based segmentation and classification procedures.

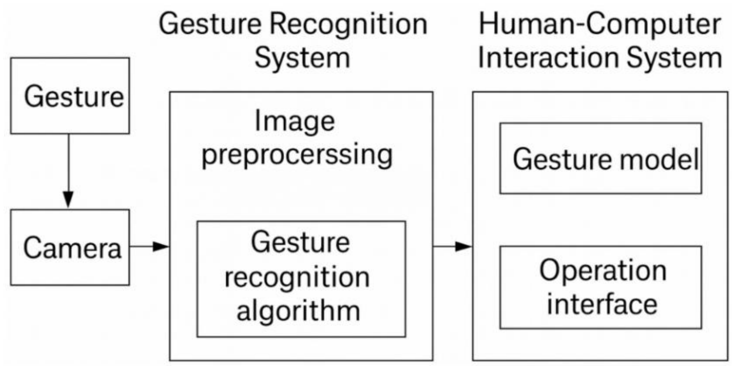

Figure 1. Computer-based gesture recognition flowchart

3. Gesture Features

3.1 GTRI Features

GTRI features are primarily used to describe image details and are typical features fre-quently used in the image recognition process. These features are based on multidimen-sional space and contain many image recognition characteristics, among which multi-angle rotation and two-dimensional spatial image scaling are representative [8-9]. Therefore, GTRI features exhibit strong adaptability to special background conditions and high light environments. They can perform comprehensive depth analysis of the target and other objects. The core working principle of GTRI is to use multidimensional space as the origin and construct the spatial quantity of GTRI features.

By calculating the spatial orientation quantities within different regions, stable feature values for each region can be obtained. During this process, the feature vectors need to be normalized. To achieve weighted calculations of image features in different regions, feature sets F and T are used. The calculation formula is expressed as:

$$F(x, y, \lambda) = H(x, y, k\lambda) - H(x, y, \lambda)$$

In the equation: $T(x, y, \lambda)$ is the feature function, k is the feature coefficient, and $H(x, y, \lambda)$ is the standard function of the feature vector. The standardized result of the feature function is used to compute the coefficients F and T , plot the path curve of the feature space, and determine the extremum of the curve as the extremum point of the feature function.

Through the above feature function, the extremum points of feature vectors can be computed. Multidimensional parameters under different combination conditions can be used to locate specific feature points. By eliminating unstable and unclear points around the image, robust feature points under challenging background and lighting conditions can be acquired. Each pixel point's location and orientation need to be determined [10-11]. Using the positions of surrounding pixels as references, the direction of the target pixel can be obtained. This enables precise localization of other feature points, forming the feature vector of GTRI, which provides a basis for image matching.

3.2 KLT Features

KLT features are commonly used for image feature region division. These features possess strong resolution and are widely applied in image processing feature algorithms [12]-[15]. KLT calculates the gray values of the image at distribution points and compares them with surrounding pixels to extract feature functions that describe image clarity. Target feature functions can be obtained through local pixel distribution. The feature function formula is expressed as:

$$T \sim t(m(g_0 - g_n) \ldots m(g_{r-1} - g_n))$$

In the equation: n represents the number of distributed feature pixels, r denotes the weight value of the spatial image pixels. When r is set to 6, 269 types of KLT feature extrema can be obtained. The KLT algorithm is shown in Figure 2.

2	6	3
7	9	5
4	1	8

(a) Numerical detection

1	1	0
1	1	0
0	1	1

(b) Normaliization

1	3	9
19	7	27
36	243	81

(c) Weight computation

Figure 2. Illustration of KLT Algorithm: (a) Numerical Detection, (b) Normalization, (c) Weight Calculation

The KLT algorithm uses a 3×3 matrix combination to form a feature function with normalized numerical values. This algorithm enables precise localization based on the spatial distribution of image pixels, effectively improving processing efficiency. However, due to the small radius of the KLT algorithm, it cannot cover most image areas, resulting in some regions being unrecognized.

To address this issue and expand the recognition range of image gray levels, researchers optimized the KLT algorithm by replacing the original 3×3 matrix with a circular region. The improved KLT algorithm can compute KLT coefficients at any arbitrary pixel within a radius R, thus generating multiple KLT computation sub-factors for labeled pixels [16]. The KLT algorithm not only offers simple implementation but also provides remark-able results in regional pixel statistics and image classification. It exhibits superior detec-tion and classification performance and demonstrates strong robustness against complex backgrounds and intense lighting environments, making it highly valuable for practical applications.

4. Gesture Segmentation Based on Edge Repair

Gesture segmentation technology involves extracting the foreground image of the gesture from the captured image and separating the hand region from the background. Accurate gesture segmentation can eliminate background interference [17]. Currently, most gesture segmentation research is based on traditional segmentation methods, including color-based skin detection segmentation, depth threshold-based segmentation, circular scan edge-based segmentation, and others. These conventional methods are often sensitive to lighting and background complexity, resulting in low segmentation speed and poor accuracy.

To address these issues, this paper proposes a gesture segmentation method based on edge repair. The overall framework of this method is shown in Figure 3. First, a rough segmentation result is obtained using skin color detection and background difference detection. Then, a hierarchical chamfer matching algorithm (HCMA) is applied to further refine the gesture region.

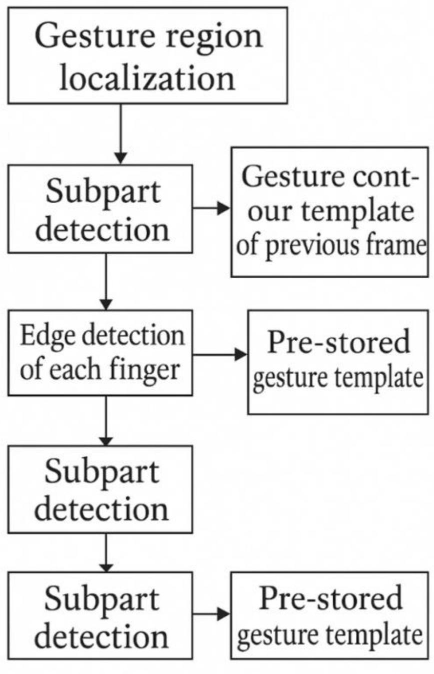

Figure 3. Framework of Edge-Repair-Based Gesture Segmentation Algorithm

Due to the flexibility of fingers, a classifier based on image structure and vertical gradient direction is used to detect the palm and fingers precisely. Finally, the boundary information at the intersection of the hand and face is repaired using pre-stored templates, resulting in a complete and clear gesture region.

4.1 Hierarchical Chamfer Distance Matching Algorithm

The Chamfer distance matching algorithm first converts the edge image into a Chamfer distance map. A 3 × 3 window is used to calculate the distance value for each pixel. The algorithm is formulated as follows:

$$Ed = \frac{1}{3}\sqrt{\frac{1}{N}\sum_{i=0}^{N-1} w_i^2}$$

In the equation, w_i represents the pixel values of the template image, and N is the total number of pixels in the template wheel [18-20]. When Ed reaches its minimum value, the best matching position is found.

To improve the search efficiency for the optimal match, the algorithm adopts a quadtree structure. Any pair of grid points is selected as the initial match position. The edge dis-tances Ed of these points and neighboring points are calculated until the minimum value is obtained. The algorithm is formulated as follows:

$$X_r = c_x + \cos\theta \cdot s_x \cdot x - \sin\theta \cdot s_y \cdot y$$
$$Y_r = c_y + \sin\theta \cdot s_x \cdot x + \cos\theta \cdot s_y \cdot y$$
$$X = \frac{X_r + 2^n - 1}{2^n}$$
$$Y = \frac{Y_r + 2^n - 1}{2^n}$$

Here, (x, y) is the template coordinate position, and (X, Y) is the coordinate on the distance map at level n of the quadtree. (c_x, c_y), (s_x, s_y), and θ represent the translation, scaling, and rotation parameters respectively, indicating that the matching process can accommodate these transformations.

4.2 Edge Repair Algorithm

The purpose of the edge repair algorithm is to obtain complete wheel shapes of gesture components. The principle is to repair broken edges. The specific steps are as follows:

(1) Use the Chamfer distance matching algorithm to search the nearby region of a gesture part and compare the best matching region image with the template for analysis.

(2) Repair the broken part of the newly detected edge image using the wheel rim break repair process. Assume a set of newly detected breakpoints P ; for each breakpoint, find its nearest point Q in the template, divide Q into multiple segments, and generate a new point set P based on their relative positions, as follows:

$$\overline{P_k} = \overline{P_{k-1}} + \frac{\overline{Q_k} - \overline{Q_{k-1}}}{2}$$

Using a Catmull-Rom interpolation curve algorithm, these points are connected to optimize the edge image [21]. The algorithm is formulated as follows:

$$
\begin{cases}
x_i' = a_i \cdot x_{k-2}' + b_i \cdot x_{k-1}' + c_i \cdot x_k' + d_i \cdot x_{k+1}' \\
y_i' = a_i \cdot y_{k-2}' + b_i \cdot y_{k-1}' + c_i \cdot y_k' + d_i \cdot y_{k+1}' \\
a_i = \frac{1}{2}(-t^3 + 2t^2 - t) \\
b_i = \frac{1}{2}(3t^3 - 5t^2 + 2) \\
c_i = \frac{1}{2}(-3t^3 + 4t^2 + t) \\
d_i = \frac{1}{2}(t^3 - t^2) \\
t = \frac{i}{N}
\end{cases}
$$

4.3 Result Analysis

To more accurately analyze gesture feature recognition in human-computer interaction, it is necessary to assign weights to the parameter factors of gesture feature recognition based on different classification methods. As shown in Table 1, in the RTF classification method, the selected parameters include feature functions, orientation node distribution, and influence factor recognition. Each parameter is assigned a weight and compared. The results show that the influence factor recognition occupies a relatively high proportion in feature classification. In contrast, the CTI classification method selects parameters such as pixel distance matching, gesture feature positioning, and algorithm accuracy checking. Using the same method, each parameter is weighted and compared. The analysis reveals that algorithm accuracy checking accounts for a larger proportion-2.5 times that of pixel distance matching. Therefore, it is necessary to analyze the segmentation accuracy of the algorithm.

Table 1: Gesture Feature Recognition Parameters

Method	Parameter	Value
RTF Classification	Feature Function Count	5
	Orientation Node Distribution	7
	Influencing Factor Recognition	13
CTI Classification	Pixel Distance Matching	51
	Gesture Feature Positioning	107
	Algorithm Accuracy Verification	126

The segmentation accuracy of the algorithm is quantitatively evaluated. 600 images were randomly selected from 15 test video sequences for manual labeling and statistical analysis. The standard testing metrics are true positive rate (TPR) and false positive rate (FPR). TPR is defined as the number of correctly predicted gesture points divided by the total number of gesture points. FPR is the number of falsely predicted background points divided by the total number of gesture points.

During recognition, if the fingers are not sufficiently extended, they may not be recog-nized. By counting the number of extended fingers, digital gestures can be identified. In addition, the meaning of a gesture can be inferred from the number of extended fingers, the name of the gesture, or the angle between fingers. Overall, the algorithm achieves fast and accurate digital gesture recognition.

Table 2: True Positive Rate (TPR) and False Positive Rate (FPR) of Gesture Segmentation Results

Method	TPR/%	FPR/%
Edge and Region-Based Segmentation	84.5	20.4
Skin Color + Background Subtraction	99	89.5
Edge-Repair-Based Component Segmentation (no sub-part)	88.7	9.4
Edge-Repair-Based Component Segmentation (with sub-part)	94.6	8.7

Based on computer vision gesture recognition research, accurate recognition and seg-mentation of gestures in images or videos are fundamental goals. Experimental results show that, using computer vision techniques, skin color segmentation combined with background difference methods, and the edge-repair-based gesture segmentation method achieve high TPR and low FPR. As shown in Table 2, the TPR and FPR of the edge-repair-based method are 94.6% and 8.7%, respectively. Compared with other methods, this approach shows superior segmentation accuracy. The edge-repair-based segmentation method offers fast recognition, high accuracy, and high research value in processing visual elements of human-computer interaction.

5. Conclusion

The edge-repair-based gesture component segmentation algorithm proposed in this paper can accurately extract the gesture region, effectively solving the problem of facial inter-ference. It also adapts to various dynamic gesture changes and has potential for real-time human-computer interaction applications.

First, the method uses GTRI features for detailed image description. Then, it applies KLT features to divide image regions. By analyzing gesture trajectory features and bone joint keypoint sequences, it achieves recognition of functions such as waving, moving, and pushing.

Compared with other algorithms, the edge-repair method achieves higher accuracy. Its advantages include low computational cost, robustness to background and distance changes, and strong suitability for long-distance dynamic gesture recognition. It exhibits good recognition stability and robustness.

However, a limitation of this method is its dependence on indoor lighting conditions. Although it shows some resistance to strong outdoor light interference, it is more suitable for indoor gesture recognition tasks and has good application value.

References

[1] M. S. Mohd Asaari, B. A. Rosdi and S. A. Suandi, "Adaptive Kalman Filter Incorporated Eigenhand (AKFIE) for real-time hand tracking system," Multimedia Tools and Applications, vol. 74, no. 21, pp. 9231-9257, 2015.

[2] X. Zhang, W. Li, X. Ye and S. Maybank, "Robust hand tracking via novel multi-cue integration," Neurocomputing, vol. 157, pp. 296-305, 2015.

[3] Y. Yu, S. Bi, Y. Mo and W. Qiu, "Real-time gesture recognition system based on Camshift algorithm and Haar-like feature," Proceedings of the 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 337-342, 2016.

[4] M. Kristan, J. Perš, S. Kovačič and A. Leonardis, "A local-motion-based probabilistic model for visual tracking," Pattern Recognition, vol. 42, no. 9, pp. 2160-2168, 2009.

[5] P. Smith, N. da Vitoria Lobo and M. Shah, "Resolving hand over face occlusion," Image and Vision Computing, vol. 25, no. 9, pp. 1432-1448, 2007.

[6] Y. Tao, M. Yang, H. Li, Y. Wu and B. Hu, "DepMSTAT: Multimodal spatio-temporal attentional transformer for depression detection," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 7, pp. 2956-2966, 2024.

[7] R. Liu, R. Zhang and S. Wang, "Graph Neural Networks for User Satisfaction Classification in Human-Computer Interaction," arXiv preprint arXiv:2511.04166, 2025.

[8] F. Dominio, M. Donadeo, G. Marin, P. Zanuttigh and G. M. Cortelazzo, "Hand gesture recognition with depth data," Proceedings of the 4th ACM/IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream, pp. 9-16, 2013.

[9] A. Núñez-Marcos, G. Azkune and I. Arganda-Carreras, "Egocentric vision-based action recognition: A survey," Neurocomputing, vol. 472, pp. 175-197, 2022.

[10] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," Proceedings of the 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, pp. 411-417, 2012.

[11] W. Liang, G. Li and H. Du, "Dynamic and combined gestures recognition based on multi-feature fusion in a complex environment," The Journal of China Universities of Posts and Telecommunications, vol. 22, no. 2, pp. 81-88, 2015.

[12] R. Jalayer, M. Jalayer, C. Orsenigo and M. Tomizuka, "A review on deep learning for vision-based hand detection, hand segmentation and hand gesture recognition in human-robot interaction," Robotics and Computer-Integrated Manufacturing, vol. 97, p. 103110, 2026.

[13] A. Mohanty, S. S. Rambhatla and R. R. Sahay, "Deep gesture: static hand gesture recognition using CNN," Proceedings of the International Conference on Computer Vision and Image Processing (CVIP 2016), vol. 2, pp. 449-461, 2016.

[14] H. Ansar, N. Al Mudawi, S. S. Alotaibi, A. Alazeb, B. I. Alabdullah, M. Alonazi and J. Park, "Hand gesture recognition for characters understanding using convex Hull landmarks and geometric features," IEEE Access, vol. 11, pp. 82065-82078, 2023.

[15] M. S. Kibbanahalli Shivalingappa, "Real-time human action and gesture recognition using skeleton joints information towards medical applications," 2020.

[16] A. Bandini and J. Zariffa, "Analysis of the hands in egocentric vision: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 6, pp. 6846-6866, 2020.

[17] M. S. Sarowar, N. E. J. Farjana, M. A. I. Khan, M. A. Mutalib, S. Islam and M. Islam, "Hand gesture recognition systems: A review of methods, datasets, and emerging trends," International Journal of Computer Applications, vol. 187, no. 2, pp. 1-33, 2025.

[18] Z. Liu, C. Zhang and Y. Tian, "3D-based deep convolutional neural network for action recognition with depth sequences," Image and Vision Computing, vol. 55, pp. 93-100, 2016.

[19] M. Yasen and S. Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications," PeerJ Computer Science, vol. 5, p. e218, 2019.

[20] M. Oudah, A. Al-Naji and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," Journal of Imaging, vol. 6, no. 8, p. 73, 2020.

[21] R. Tripathi and B. Verma, "Survey on vision-based dynamic hand gesture recognition," The Visual Computer, vol. 40, no. 9, pp. 6171-6199, 2024.