



OPEN Hand gesture 3D pose estimation method based on swin transformer and CNN

Rong Dang^{1,2✉} & Gang Feng¹

Existing gesture pose estimation methods commonly exhibit limitations such as singular feature extraction for hand characteristics and the neglect of long-range topological relationships between joints, thus restricting their prediction accuracy. To address these issues, this study proposes a gesture pose estimation method that takes depth images as input. First, rough gesture features are extracted using a convolutional network, while a Swin Transformer module captures the topological relationships between joints and global features through spatial information. A U-shaped network is used to process the features hierarchically, preserving local joint information at various resolutions, which is fused with global features. Two-dimensional Gaussian heatmaps are introduced to represent the distribution of keypoints, improving network supervision for target feature regression. The backend network outputs the final keypoint coordinates. We evaluated the method on a newly constructed dataset, achieving a reduced average squared error of 7.012–4.776 mm lower than that of the baseline model. The experimental results demonstrate the superior performance of the proposed method, when comparing it to state-of-the-art mainstream pose estimation networks.

Natural and intuitive gesture-based interaction is an essential approach for enhancing human–computer interaction experiences, with broad applications in areas such as virtual reality (VR), augmented reality (AR), smart cockpits, sign language recognition, and remote collaboration^{1–4}. Hand pose estimation refers to the process of recovering the three-dimensional spatial positions of hand joints from input visual data, such as RGB images or depth maps. Specifically, the goal of pose estimation is to obtain accurate information about the position and orientation of the hand using computer vision methods, which is essential for natural human–computer interaction⁵. For example, by employing hand pose estimation, users can interact with objects in virtual environments through gestures or perform tasks such as sign language recognition⁶. The key to enabling these applications lies in robust and accurate hand pose estimation techniques, whose objective is to recover the 3D coordinates of predefined hand keypoints from input visual data (e.g., RGB images and depth maps). Such techniques allow users to interact with computers and devices in a natural and intuitive manner, thereby achieving a more intelligent, convenient, and user-friendly interaction experience⁷.

For example, Oikonomidis et al. proposed a multi-view method that employs a particle swarm optimization algorithm to minimize the discrepancy between projected 2D hand model features and those extracted from input sequences⁸. This approach effectively incorporates physical constraints as critical reference information, significantly improving estimation accuracy under occluded background conditions. Furthermore, with the remarkable success of deep learning in human–computer interaction, Tompson et al. applied CNNs to hand pose estimation⁹. By generating 2D heatmaps through convolutional neural networks and leveraging inverse kinematics, they predicted the coordinates of hand joints. Ye et al. further introduced a hand pose estimation method based on a kinematic hierarchical strategy¹⁰. Their approach integrates the proposed strategy into the input space of a spatial attention-based discriminative method and the optimization process of a hierarchical particle swarm optimization-based generative method, thereby improving the accuracy of pose calibration to some extent.

Although significant progress has been made in the field of Hand pose estimation, existing methods still face several common challenges: (1) the extracted features are often limited in diversity and lack effective fusion of multi-scale gesture information, ranging from global structures to local details; and (2) the intrinsic spatial topology of the hand, particularly the long-range dependencies between non-adjacent joints (e.g., collaborative motion constraints among fingers), is not fully exploited. These issues directly hinder further improvement in pose estimation accuracy.

¹School of Architecture, Tianjin University, Tianjin 300073, China. ²China Construction Engineering Design and Research Institute Co., Ltd, Beijing 100045, China. ✉email: drong_7788@tju.edu.cn

To address these challenges, this paper proposes a hand pose estimation method based on Swin transformer and CNN, which aims to simultaneously overcome the limitations of single-feature extraction and the neglect of topological dependencies. The main contributions are as follows:

- Coarse hand gesture features are first extracted using a convolutional network and then passed into the Swin Transformer module, which focuses on the topological dependencies of hand joints. This design aims to capture global features enriched with spatial information under attention computation.
- A U-shaped network is designed, where residual structures are employed to propagate hierarchical information, enabling the extraction of multi-scale features of local hand joints at different resolutions. These local features are fused with global features. In addition, a two-dimensional Gaussian heatmap encoding the probabilistic distribution of hand joint locations is introduced to better supervise the network in regressing target features.
- Finally, high-quality and diverse depth-map-based 3D hand pose data are collected and annotated using LiDAR devices. Together with the MSRA dataset¹¹, these data are integrated to construct a new experimental dataset.

The remainder of this paper is organized as follows: Sect. "Related work" introduces relevant work research. Section "Methods" provides a detailed description of the proposed model for 3D hand pose estimation. Section "Experiments and analysis of results" presents the experimental setup and analyzes the results. Finally, Sect. "Conclusions" concludes the paper and outlines future research directions.

Related work

At present, from the perspective of gesture modeling, hand pose estimation methods can be categorized into three types: generative model-based methods, discriminative model-based methods, and hybrid model-based methods¹².

Generative model-based estimation methods typically use optimization algorithms to combine the motion trajectory of the hand with predictions from a dynamics perspective, followed by the extraction of inputs, integrating shape features of the model to minimize matching errors and obtain precise pose parameters¹³. Qian et al. improved this method by constructing a geometric model of the hand¹⁴. They adopted the iterative closest point (ICP) model to accelerate convergence through local optimization and utilized the particle swarm optimization (PSO) algorithm for stochastic optimization. By expanding the search in the parameter space, they effectively avoided becoming trapped in invalid local optima. Zhang et al. employed a multi-task learning framework to simultaneously estimate a three-dimensional (3D) hand pose, a hand mask, and a mesh-based anthropomorphic hand outline (MANO) mesh¹⁵. Chen et al. proposed I2UVHandNet, which unfolds the low-resolution MANO 3D hand mesh into a 2D UV plane, enhances its resolution through a super-resolution network layer, and then reconstructs it into a high-resolution MANO mesh¹⁶. However, such generative methods require the construction of intricate hand models and combination of deep network models to find the most similar pose parameters within a consistent time frame, making real-time performance a challenge.

On the other hand, the core idea of discriminative-based estimation methods is to directly infer the positions or poses of hand joints from single-frame images or depth data using machine learning or deep learning models. These methods do not rely on predefined hand models or templates, instead learning the mapping relationship from input to output directly via training on large amounts of annotated data. Ge et al.¹⁷ proposed an end-to-end trainable hand mesh generation method using graph convolutional networks. Zhang et al. proposed an end-to-end hand pose estimation method consisting of three parts: the first part estimates the 2D heatmaps of hand joints, the second part iteratively regresses the hand model parameters using these heatmaps as input, and the third part generates the corresponding hand model based on these parameters¹⁸. To address the challenge of balancing accuracy and efficiency in hand pose estimation, Fan et al. introduced the adaptive computationally efficient (ACE) network, which dynamically switches between lightweight and complex networks using a Gaussian kernel-based gating module¹⁹. Pavlakos et al. presented HaMeR (hand mesh recovery), a method that achieves 3D hand reconstruction from a monocular input. In particular, it employs a Transformer-based architecture that significantly enhances the accuracy and robustness of hand analysis compared to existing technologies²⁰.

Rezaei²¹ introduced TriHorn-Net, a novel model that decouples the 3D hand pose estimation task into two sub-tasks—namely, estimating the 2D joint positions (UV coordinates) in the depth image space and estimating the corresponding depth values aided by two complementary attention maps—significantly improving the accuracy of hand pose estimation based on depth images. Wen et al.²² developed a Transformer-based temporal modeling framework, constructing a cascaded dual Transformer encoder hierarchical network that utilizes short-term temporal cues for effective hand pose estimation in the first encoder and integrates per-frame pose information with object interaction features for the recognition of actions over longer time spans.

However, as only 2D positions are predicted, the performance of such methods in scenarios with joint occlusion is sub-optimal. Hybrid-based estimation methods primarily treat the constructed hand model as a prerequisite for generating hand coordinates or as a discriminative condition for regression targets. For example, Keskin et al.²³ conducted experiments on synthetic depth images, attempting to use a local mode search algorithm to predict hand joint coordinates from different results. Sridhar et al.²⁴ introduced an innovative detection-guided optimization strategy that enhances the robustness and speed of pose estimation. Later, Sridhar et al.²⁵ presented a 3D joint Gaussian mixture alignment strategy for hand pose estimation that is suitable for rapid pose optimization, employing a regularizer to address occlusion issues arising from hand–object contact. Panteleris et al.²⁶ proposed a three-step approach for hand pose estimation: first, an existing and effective hand detector is

used to localize and segment the hand; second, the 2D positions of hand joints are estimated using a pre-trained OpenPose discriminative network model; finally, the 2D pose is fitted to a 3D pose using nonlinear least squares.

Although existing methods have made progress, they generally suffer from limited feature representation and insufficient global topology modeling. The method proposed in this paper, which integrates Swin transformer with CNN, is designed to address these shortcomings.

Methods

The overall design of the proposed model is illustrated in Fig. 1, and the model design involves multiple steps for gesture feature extraction. First, the depth image of the hand region is input into a convolutional feature extraction network, where rough gesture features are extracted through a 7×7 convolution layer and further optimized using a pooling unit and bottleneck residual modules. Next, the features are processed in parallel through two branches: one branch employs a U-Net to obtain local joint features, while the other branch uses a Swin Transformer encoder to capture global features. The output feature maps from both branches are fused, and a Gaussian heatmap containing joint distribution information is introduced to supervise and correct the extracted features. This process is repeated twice to further refine the extracted gesture features, ultimately resulting in accurate three-dimensional joint coordinates.

Global topological information extraction

To better understand the relationships between hand joints in the task of hand pose estimation (i.e., their relative positions and connections in space), this study introduces a Swin Transformer to extract the topological information of hand joints. Compared to the traditional Vision Transformer mechanism, the Swin Transformer incorporates a window-based multi-head self-attention (W-MSA), which divides the input features into smaller patches. Through the use of multi-level attention mechanisms and cross-layer fusion, the model can more efficiently process long-range dependency information, integrating local hand feature information captured at different levels into a global representation. This structure also reduces the computational complexity, to a certain extent. The specific architecture of the Swin Transformer encoder is illustrated in Fig. 2.

Figure 2 illustrates the specific structure of the Swin Transformer encoder. The first part is the W-MSA, while the second part is the shifted window-based multi-head self-attention (SW-MSA). Each component is composed of a layer normalization layer, an attention module, and a multi-layer perceptron (MLP). The overall structure includes an input section, a patch partition block, and four stages for extracting hand gesture features. The Swin Transformer effectively captures the long-range dependencies among hand joints through its windowed multi-head self-attention mechanism. Input features are divided into smaller blocks (windows), with self-attention calculated within each window, significantly reducing computational costs. The shifted window mechanism allows for the transfer of information across layers, facilitating the effective fusion of hand joint features at different scales. This multi-scale attention is crucial for accurate gesture pose estimation. The specific process is outlined as follows:

First, a feature map of size 48×48 is fed into the Swin Transformer Block, with the window size set to 4×4 . At this point, the image is divided into 12×12 windows and self-attention computation is performed within each window, converting the regional pixels into one-dimensional vectors to be used as the effective input sequences. Then, residual connections are introduced through the MLP for linear embedding, aiming to project the features according to their dimensions. As shown in Fig. 2, after passing through the W-MSA in the block, the resulting hand feature map undergoes another SW-MSA structure, where the window positions are shifted by 2 pixels in both the x and y directions to enable the exchange of information between different windows.

The remaining three stages adjust the size and channel count of the feature map through patch merging operations, while the Swin Transformer block module performs a series of linear transformations and attention

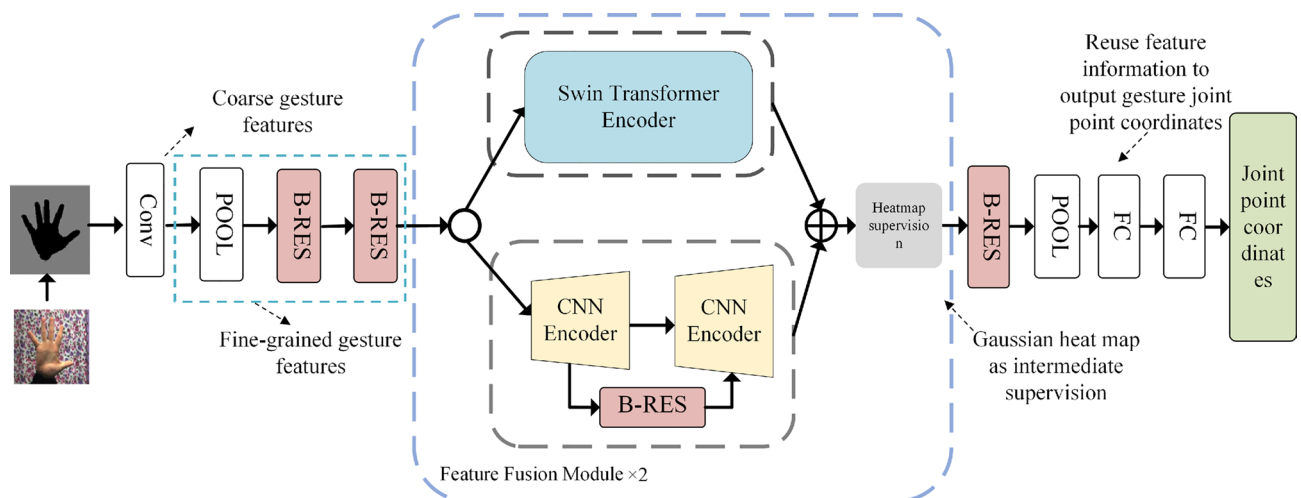


Figure 1. The overall gesture estimation network structure.

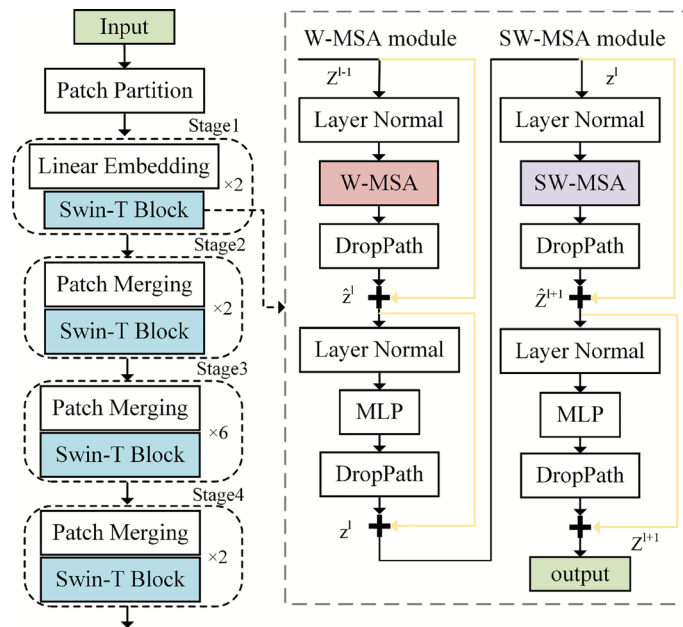


Figure 2. Design of the swin transformer encoder-decoder structure.

mechanism operations. Each stage modifies the tensor dimensions to capture hand features at different levels, aiding the model in globally understanding the feature information. Finally, the output is processed through a layer normalization layer to maintain consistency in the feature distribution and obtain the corresponding results. Let $\hat{z}^l$ and z^{l+1} represent the window output features, and $\hat{z}^l$ and z^{l+1} represent the MLP output features. The computational process of the Swin Transformer block can be expressed by Eqs. (1), (2), (3), and (4):

$$\hat{z}^l = W - MSA(LN(z^{l-1})) + z^{l-1}, \quad (1)$$

$$z^l = MLP(LN(\hat{z}^l)) + \hat{z}^l, \quad (2)$$

$$\hat{z}^{l+1} = SW - MSA(LN(z^l)) + z^l, \quad (3)$$

$$z^{l+1} = MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1}. \quad (4)$$

Additionally, before the final output layer, the Swin Transformer generates feature maps at different levels, encompassing multi-scale features that range from fine- to coarse-grained. In standard Transformers, the computational complexity of the self-attention mechanism grows quadratically with increasing sequence length. For images, directly treating individual pixels as a sequence can lead to a significant computational burden. The Swin Transformer mitigates this issue by partitioning the image into multiple non-overlapping local windows and calculating the self-attention within these windows independently, which significantly reduces computational costs.

To facilitate information exchange between different windows and capture a broader context, the shifted window mechanism is introduced. Between successive layers, the positions of the windows shift, resulting in overlapping portions between adjacent layers. This design not only maintains a lower computational load but also allows for the effective integration of information across windows. Following these principles, the Swin Transformer captures subtle features and spatial dependencies in gesture pose estimation images through block-based processing, self-attention mechanisms, and multi-scale hierarchical modeling, providing powerful modeling capabilities for gesture pose estimation.

Local joint feature information aggregation

In some image processing tasks, neural networks are stacked to gradually extract feature information. In shallow networks, the model primarily learns low-level features. As the network deepens, the information learned by the network becomes more abstract as the resolution of the feature maps decreases. However, for tasks such as hand pose estimation, which require the extraction of fine-grained features, the key information for different joints may exist in feature maps of varying resolutions. Therefore, by aggregating these diverse features distributed across different resolutions, more accurate information can be captured to guide the network's learning process. Inspired by the hourglass network proposed by Newell's team^{27–29}, this study designs a U-shaped encoder-decoder to capture multi-scale feature information from feature maps with different resolutions through progressive down-sampling and up-sampling operations. Figure 3 illustrates the structure of the U-shaped encoder-decoder.

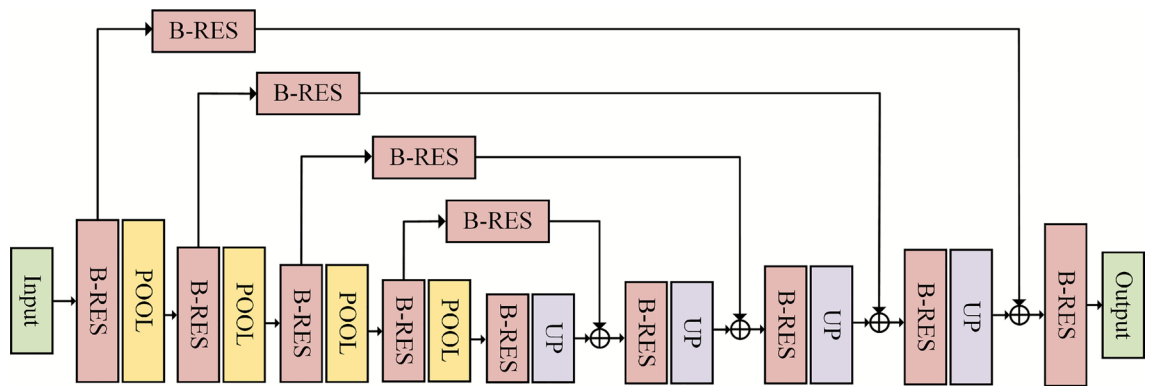


Figure 3. U-shaped network structure design.

Figure 3 illustrates the structural design of the U-Net, which extracts and aggregates multi-scale features from feature maps of varying resolutions through downsampling and upsampling operations. Among the components of this network, B-RE represents the bottleneck residual structure, POOL represents the max pooling structure, and UP represents the up-sampling structure. The introduction of the bottleneck residual module and pooling layer enables the network to learn detailed features of the gestures at different scales. The upsampling phase restores the resolution of the feature maps and combines low-level gesture features at the same resolution with high-level features resulting from residual processing via skip connections. This multi-resolution feature aggregation process is crucial for accurate gesture pose estimation.

In the top-down process, the bottleneck residual structure and the max pooling structure appear in groups. The network starts from a high-resolution feature map and reduces the resolution by a factor of 2. At each step, the detailed information of the features is enhanced and the number of feature channels doubles accordingly. In the bottom-up process, the bottleneck residual structure and the up-sampling structure appear in groups. In this part, the output result of each previous layer needs to be up-sampled by a factor of 2, with the aim of restoring the original resolution of the feature map. In this step, it is also necessary to add—through a skip connection—the low-level gesture features processed by the B-RES module at the same resolution.

At the same time, the network starts from a low-resolution feature map and increases exponentially. The number of feature channels also decreases by a factor of 2 until it finally returns to the original state. At this point, the obtained features are the gesture feature information aggregated from four different resolutions. The up-sampling method adopted in this study is the bilinear interpolation method. Through considering the weights of the four adjacent pixels, the value of the target pixel is estimated and each pixel in the input feature map is interpolated to generate a feature map with a larger size, thus achieving smooth up-sampling of the image and restoring the original information of the feature map to a large extent.

The bottleneck residual module (B-RES) is a type of module used for constructing deep neural networks in the field of deep learning, which is an improvement on the basis of the traditional residual network (ResNet) structure. By adding a bottleneck layer, it introduces more depth and nonlinear transformations into the network, enabling the network to better adapt for the extraction of multi-scale features and endowing it with greater flexibility. In the U-shaped network, the number of feature channels also exhibits a U-shaped trend, increasing first and then decreasing, and the obtained features also differ. Introducing the bottleneck residual structure helps the network to better transmit residual information: it not only preserves and transmits global and local information, but also captures and aggregates the hand feature information at different scales. This structure can also reduce the computational complexity to a certain extent, making the network lighter and more suitable for practical deployment.

Regarding multi-resolution feature maps, the bottleneck residual module can play a more effective role in gesture feature extraction when compared to the ordinary residual module. A comparison between the bottleneck residual and traditional structures is shown in Fig. 4, where Conv represents the convolution operation constructed through 1×1 , 3×3 , and 1×1 convolution kernels. This effectively reduces the number of parameters in the network, making the model lighter and more capable of handling multi-scale features. The bottleneck residual structure reduces the number of input feature channels in the first 1×1 convolutional layer, then conducts more in-depth feature extraction in the second 3×3 convolutional layer, and finally restores the depth of the features in the third 1×1 convolutional layer.

Therefore, it can extract richer and more complex feature information while maintaining the size of the feature map. In addition, when considering multi-resolution feature maps, the effect of gradient propagation is particularly important. When extracting features at different scales, the stable propagation of gradients can help the network to better learn the feature representations at different scales. The bottleneck residual structure directly transmits the input to the output end through the residual connection, which helps to alleviate the problem of gradient vanishing and makes the network easier to train.

Gesture heatmap supervision

In the task of gesture pose estimation, the three-dimensional information representing the positions of the key points of the hand joints ultimately needs to be obtained through the established model. During this process,

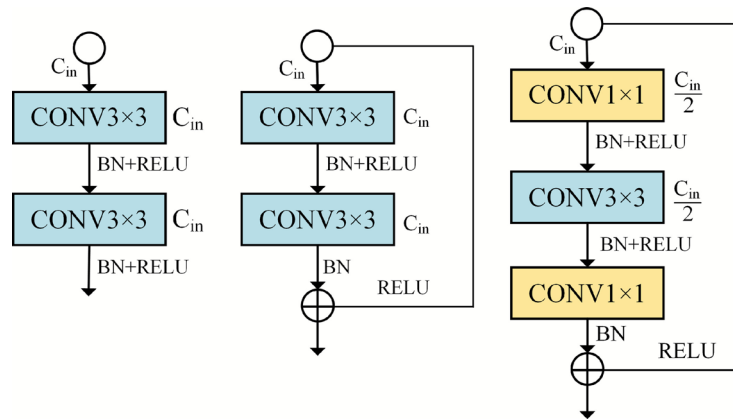


Figure 4. Comparison of residual structures.

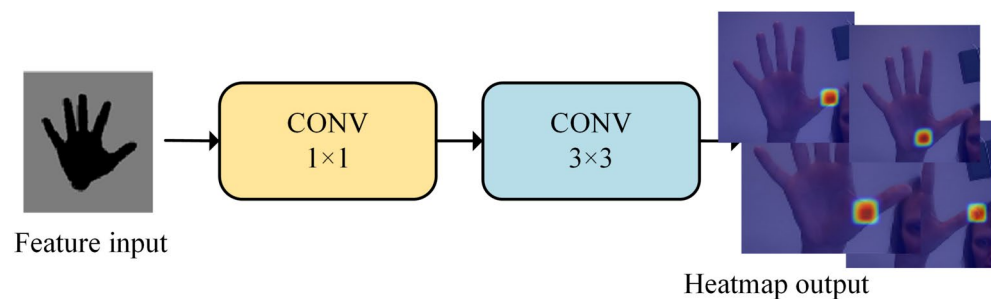


Figure 5. Example of heatmap-based supervision module operation.

a fully differentiable model is often used for training. The error between the input data and the target value can be directly utilized to optimize the model parameters, which simplifies the training process and improves the training efficiency. Due to this end-to-end weight update mode, the forward and backward propagation speeds are relatively fast. However, especially for tasks such as gesture pose estimation, there may be significant differences in the gesture shapes and postures between different users and in different environments. Therefore, the lack of spatial generalization ability may cause the model to perform poorly in practical applications.

Inspired by reference³⁰, in this study, a two-dimensional Gaussian distribution centered on the positions of the points representing the hand gesture joints is generated. The variance of the Gaussian distribution is usually controlled using pre-set parameters. The pixel with the highest intensity value indicates the position of the key point, and the intensity values of the surrounding pixels gradually decrease. By superimposing the Gaussian distributions generated for all key points, the heatmap value for each pixel can be obtained. Therefore, by finding the position with the highest confidence value in the heatmap, the key points in the hand gesture can be more accurately located. At the same time, the topological connections between the joint points are also retained according to the peaks in the distribution. In this way, the model can process multiple key points simultaneously, enabling the model to better understand the structure and dynamics of the hand gesture and make more accurate predictions.

The structural design of the heatmap supervision module is illustrated in Fig. 5. The input is a two-dimensional matrix containing the distribution information of keypoints, where the intensity value of each pixel represents the confidence of the keypoint. The role of the convolutional kernel is to slide over the keypoint heatmap, using a 1×1 convolution to increase the number of input feature channels to 256. The learned weights are then used to identify and capture the patterns and features of the gesture keypoints. Subsequently, a 3×3 convolution operation processes the features obtained through the 1×1 convolution, outputting a Gaussian heatmap of size 24×24 with a number of channels corresponding to the number of gesture keypoints.

This extracts the patterns and features of the gesture keypoints from the input features. For each key point u , there exists a term $u \in R^2$. At this time, we can calculate $\tilde{u} = \frac{u}{R}$. All ground-truth keypoints are then decomposed into a heatmap, denoted as $H \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times c}$. The mathematical representation of the Gaussian heatmap is given by Eq. (5):

$$H_n(u, v) = \exp \left(-\frac{(u - u_n)^2 + (v - v_n)^2}{2\sigma^2} \right), \quad (5)$$

where (u_n, v_n) represents the joint position coordinates, and σ is the standard deviation of the Gaussian function. The value is maximal at the joint's position, and gradually decreases as the distance from the joint position increases.

Loss function

The loss function adopted in this study follows a common approach used in gesture pose estimation methods, consisting of two parts: heatmap regression loss and joint point regression loss.

Heatmap regression loss: The mean squared error is calculated between the Gaussian heatmap generated from the ground-truth information of the image and the Gaussian heatmap produced according to the model's predicted joint points. The calculation formula is shown in Eq. 6:

$$L_{ht} = \sum_{m=1}^M \sum_{n=1}^N \sum_{u,v} \| H_n^{m*}(u, v) - H_n^m(u, v) \|^2, \quad (6)$$

where $H_n^{m*}(u, v)$ represents the m th Gaussian heatmap generated by the model at the position of the n th joint point, $H_n^m(u, v)$ denotes the m th Gaussian heatmap produced from the ground-truth information at the position of the n th joint point, N represents the total number of joint points in the detected hand region, and M represents the stacked fused features.

Joint point regression loss: The mean squared error is calculated between the annotated ground truth joint point coordinates and the joint point coordinates predicted by the network model. The calculation formula is shown in Eq. 7:

$$L_j = \sum_{n=1}^N \| J_n^* - J_n \|^2, \quad (7)$$

where J_n^* represents the predicted coordinates of the n th joint point and J_n denotes the coordinates of the n th joint point in the annotated gesture. The overall loss is obtained by weighting these two loss functions, calculated as shown in Eq. 8:

$$L = \alpha \times L_{ht} + L_j, \quad (8)$$

where α represents the weight value, L_{ht} represents the heatmap regression loss, and L_j represents the joint point regression loss.

Experiments and analysis of results

This section provides a detailed explanation of the experiments and results analysis conducted to validate the proposed method. Specifically, Subsect. [Datasets](#) describes the datasets used in the experiments, Subsect. [Evaluation metrics](#) outlines the evaluation metrics, Subsect. [Experimental settings](#) presents the experimental environment setup, Subsect. [Comparative experiments with existing methods](#) provides a comprehensive analysis of the experiments, and Subsect. [Ablation experiments](#) details the ablation experiments of the proposed method.

Datasets

The experimental dataset used in this study comprises two parts, as illustrated in Fig. 6. On the right side is the MSRA dataset, which was constructed by Microsoft Research Asia. It includes 76,375 depth images generated from nine subjects presenting diverse gestures under varying viewpoints and lighting conditions, with each image accurately annotated for the 3D coordinates of 21 hand joints.

The scientific design of this dataset is evident in its rigorous collection protocol: all gesture actions encompass typical postures used in daily interactions, including gripping, pinching, and spreading, while accounting for variations in hand size and skin color to ensure the representativeness and diversity of the data. As shown



Figure 6. Examples from the LiDAR and MSRA datasets.

Dataset	Data volume	Annotation dimension	Applicable tasks	Features
MSRA	76k	3D	Hand pose estimation	High-precision depth data
RHD	44k	2D/3D	Synthetic data research	Synthetic data, no real noise
FreiHAND	130k	3D	Real-time hand gesture estimation	Contains dynamic sequences

Table 1. Comparison results between MSRA and other datasets.

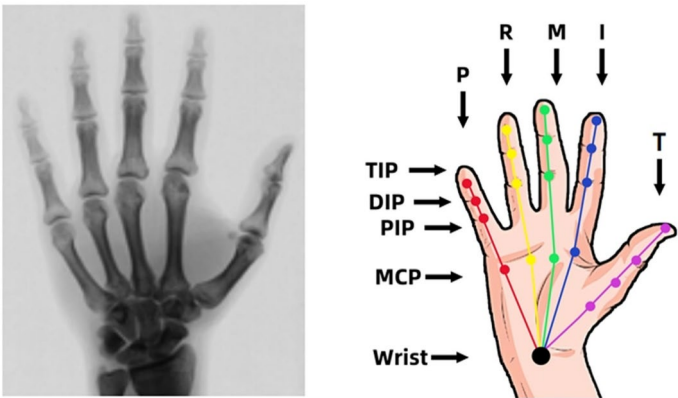


Figure 7. Annotation of the 21 joints of the hand and the finger labels.

in the table below, the MSRA dataset is the preferred benchmark for validating the performance of 3D hand pose estimation algorithms due to its real-world collection environment (non-synthetic data), high-precision annotations (sub-millimeter accuracy), and extensive number of samples.

Additionally, the depth information contained in the dataset makes it particularly well-suited for studying challenging issues in real-world scenarios, such as occlusion and lighting changes, providing an important experimental platform for developing effective hand interaction systems. These characteristics collectively form a solid scientific basis for the use of the MSRA dataset. The results of a comparative analysis between the MSRA dataset and other datasets are presented in Table 1.

On the left side of Fig. 6, self-collected LiDAR data are shown, which include data for eight types of gestures which we self-captured using a CS30 solid-state LiDAR sensor. Furthermore, a 21-joint labeling method was employed to annotate these gestures. The primary goal of including these data was to address the lack of long-distance and small-target samples in the standard MSRA dataset, thus improving the robustness and generalization performance of the proposed model.

To ensure clarity and simplicity in the subsequent experimental data labels, the 21-joint annotation information was simplified as follows. The five fingers of the hand—namely, the little finger, ring finger, middle finger, index finger, and thumb—are denoted as P, R, M, I, and T, respectively. Each finger is represented by a fingertip (TIP) and three other joints: DIP, PIP, and MCP. The center of the wrist is represented by Wrist. For ease of explanation in the experimental charts, TIP, DIP, PIP, and MCP are further simplified to the numbers 1–4, while Wrist is represented by C. Figure 7 illustrates the hand annotation information.

Evaluation metrics

This study employs two commonly used evaluation metrics for hand 3D pose estimation: the mean per joint position error (MPJPE) and percentage of correct keypoints (PCK), with a threshold range from 0 to 80 mm, in order to assess the effectiveness of the proposed method. In particular, PCK refers to the percentage of correctly predicted 3D joints.

(1) MPJPE is calculated as the average Euclidean distance between the joint coordinates predicted by the model and the ground-truth coordinates, expressed in millimeters (mm). The calculation formula is shown in Eq. (9):

$$\text{MPJPE} = \frac{1}{T} \frac{1}{N} \sum_{t=1}^T \sum_{i=1}^N \|J_i^{(t)} - \hat{J}_i^{(t)}\|_2, \tag{9}$$

where MPJPE represents the mean per joint position error; T denotes the number of samples; N represents the number of joint points; $J_i^{(t)}$ represents the position information of the i th joint in the annotation of the t th image; and $\hat{J}_i^{(t)}$ represents the predicted position information of the i th joint in the t th image.

(2) PCK is defined as the percentage of predicted keypoints that fall within a given distance threshold relative to the ground-truth joint coordinates. The calculation formula is shown in Eq. (10):

$$PCK_i^k = \frac{\sum \delta \left(\frac{d_i}{d^{def}} \leq T_k \right)}{\sum_k 1} s, \tag{10}$$

where T_K is the threshold, d_i is the Euclidean distance between the predicted value and the ground-truth of the i th joint point, and d^{def} is the normalization factor.

Experimental settings

The experiments in this study were conducted on the Ubuntu 16.04 operating system, with a hardware environment consisting of an Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz, 40 GB of GPU memory, and an RTX 3080 GPU. The deep learning framework used was PyTorch 1.7.1 with CUDA version 10.2. The initial learning rate for the model was set to 0.001 and the momentum was set to 0.9. The total number of training epochs was 300, with a batch size of 64. The standard deviation of the Gaussian function σ was set to 2, and the weight in the loss function α was set to 1. All experiments utilized the Adam optimizer with default hyperparameters, and training, validation, and testing were conducted under the same hyperparameter settings.

Comparative experiments with existing methods

To evaluate the performance of the proposed hand pose estimation algorithm, a series of comparative experiments were carried out. Five representative mainstream hand pose estimation algorithms—namely, DeepModel³¹, REN³², DeepPrior++³³, PoseREN³⁴, and DenseReg³⁵—were selected as benchmarks. These algorithms have all demonstrated excellent performance in their respective fields, providing a solid reference basis for this comparative experiment. The experiment focused on two key performance indicators: the average error of the gesture joint points and the average prediction frame rate. Through comparison in these two dimensions, we can intuitively understand the advantages of the algorithm proposed in this paper compared with other mainstream methods. The experimental results are presented in Table 2.

Through comparison of the data, it can be observed that: on the dataset used in this paper, compared with the DeepModel, REN, DeepPrior++, PoseREN, and DenseReg algorithms, the MPJPE value of the method proposed in this paper was significantly lower than that of the other mainstream gesture pose estimation algorithms; in particular, compared with the DeepModel, REN, DeepPrior++, PoseREN, and DenseReg algorithms, the average joint point error of the proposed algorithm was reduced by 1.655 mm, 0.638 mm, 0.871 mm, 1.605 mm, and 0.206 mm, respectively. Moreover, compared with the DenseReg algorithm, the average joint point error of the proposed algorithm was reduced by approximately 2.9%.

This result directly proves the high precision of the proposed method in predicting the positions of hand joint points, effectively reducing the estimation error. Although the proposed method was slightly inferior to the DeepModel and REN models in terms of the average prediction frame rate, this gap remains within a reasonable and acceptable range. While ensuring high accuracy, the proposed algorithm does not sacrifice excessive computational efficiency and can still meet the requirements of real-time or near-real-time applications. Therefore, considering the model performance in terms of both accuracy and real-time capability, the gesture pose estimation method proposed in this paper demonstrates significant advantages.

To more fully illustrate the prediction accuracy of our method, Fig. 8 presents a detailed performance comparison of our proposed approach with respect to the DeepModel, REN, DeepPrior++, PoseREN, and DenseReg algorithms within the 0–80 mm range of PCK. In the graph, the x-axis represents the allowable offset error for the estimation of hand key points, while the y-axis indicates the accuracy of the 3D key point estimations within the error range, allowing for calculation of the area under the curve to assess the performance of each model. As shown in Fig. 8, the overall PCK curve aligns with the data presented in the previous table. The proposed method maintains a high PCK value within the 0–80 mm error threshold range, with its advantage becoming increasingly pronounced as the threshold expands.

Notably, within the 0–15 mm low-error threshold range, the DenseReg and PoseREN methods exhibit a slight advantage over the proposed method in certain areas. This is due to these methods’ more direct optimization of local feature regression. However, within the broader and more practically significant 15–80 mm error threshold range, the proposed method consistently and significantly outperforms the other baseline methods. This phenomenon indicates that the global topological relationships modeled by the Swin Transformer and the multi-scale local features fused by the U-Net enable the model to maintain higher accuracy in the face of challenges such as gesture variation and partial occlusion, with smaller deviations from the true values. As a

Algorithm	Average joint point error (mm)	Average prediction frame rate (fps)
DeepModel ³¹	8.676	215
REN ³²	7.659	48
DeepPrior++ ³³	7.892	30
PoseREN ³⁴	8.626	36.7
DenseReg ³⁵	7.227	27.8
Ours	7.021	38

Table 2. Performance comparison of different algorithms.

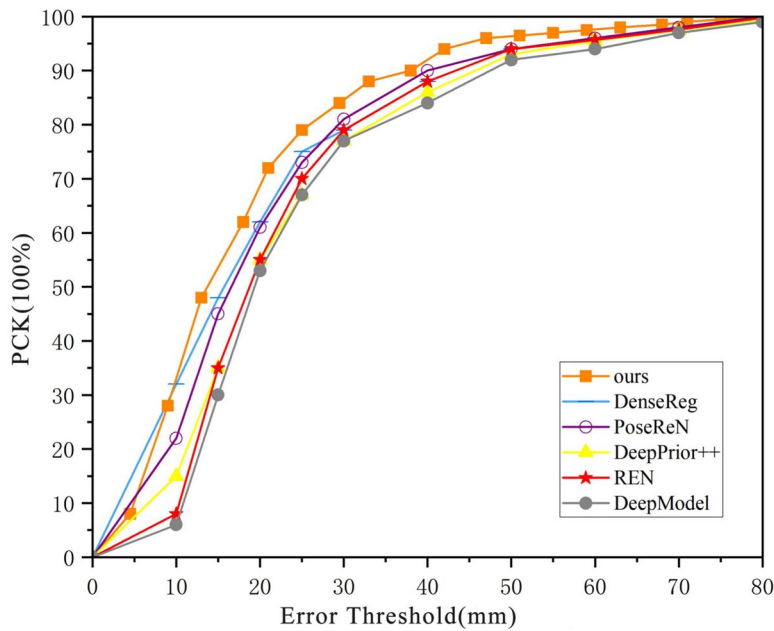


Figure 8. Comparison of PCK curves for different methods.

Number	Model scheme	Mean squared error of all joints (mm)
1	Baseline	15.698
2	Baseline +A	12.357
3	Baseline +B	13.596
4	Baseline +A+B	11.157
5	Baseline +A+B+C	10.922

Table 3. Ablation experiment results.

result, the proposed method demonstrates a core competitive advantage in maintaining higher accuracy within a more lenient tolerance range.

Ablation experiments

To comprehensively evaluate the contribution of each key module in the proposed method to its hand joint point estimation accuracy, a series of ablation experiments were performed. These experiments revolve around three core modules: The global feature extraction module (A), the multi-resolution local feature aggregation module (B), and the heatmap supervision module (C). Additionally, the baseline mentioned in the experiment consists of a basic convolutional network, which includes convolutional layers, max-pooling layers, and several bottleneck residual modules for feature extraction. A regression head is subsequently applied to directly output the 3D coordinates of the hand keypoints, without incorporating Modules A, B, or C. The proposed method was adjusted into different network structures for testing. To ensure the accuracy of the experiments, the same parameters were used during the training process, and the experiments were all carried out on the same self-made dataset. The results of the ablation experiments are shown in Table 3.

It can be seen from the table that each key module in the proposed method significantly contributed to improving the accuracy of hand joint point estimation. Specifically, the global feature extraction module and the multi-resolution local feature module both reduced the error estimation of hand joint points to a certain extent, when compared the benchmark model. Moreover, when both the global feature extraction module and the multi-resolution local feature module were used, the resulting error value was the lowest, having been reduced by 4.541, thus demonstrating the effectiveness of the proposed model for gesture pose estimation. In addition, with the heatmap supervision module, the background in the image compared with the predicted joint points becomes clearer, guiding the network to learn autonomously and further reducing the joint point prediction error, which demonstrates the effectiveness of this approach.

Figure 9 shows a histogram of the average error of each joint point in the ablation experiment. In this way, the differences in the performance of the model under different configurations can be clearly seen.

To more clearly demonstrate the performance advantages of our proposed method in real-world applications, we performed a visual comparison of the network prediction results on both the LiDAR dataset and the MSRA benchmark dataset. Figure 10 presents the visual results from our custom LiDAR dataset, where green represents

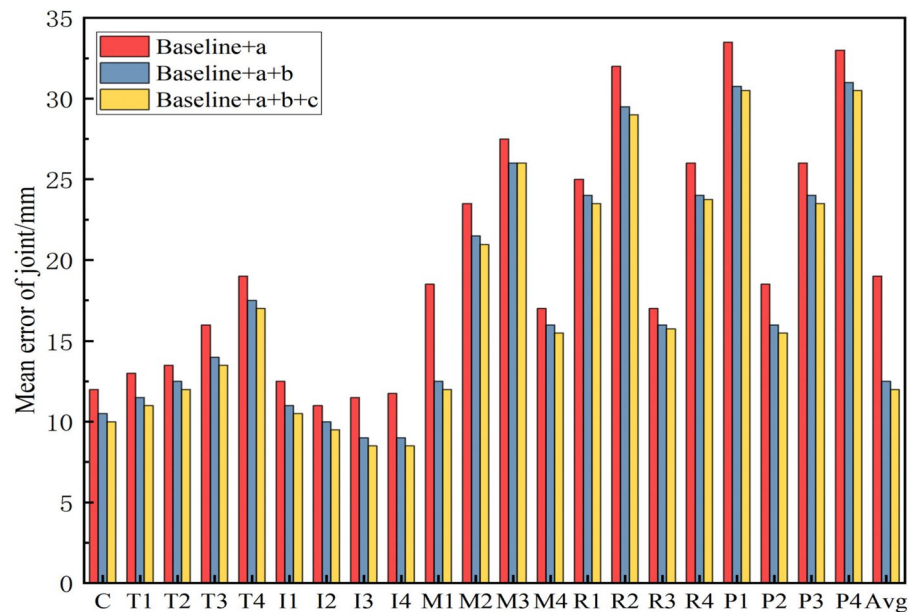


Figure 9. Results of the ablation experiment.

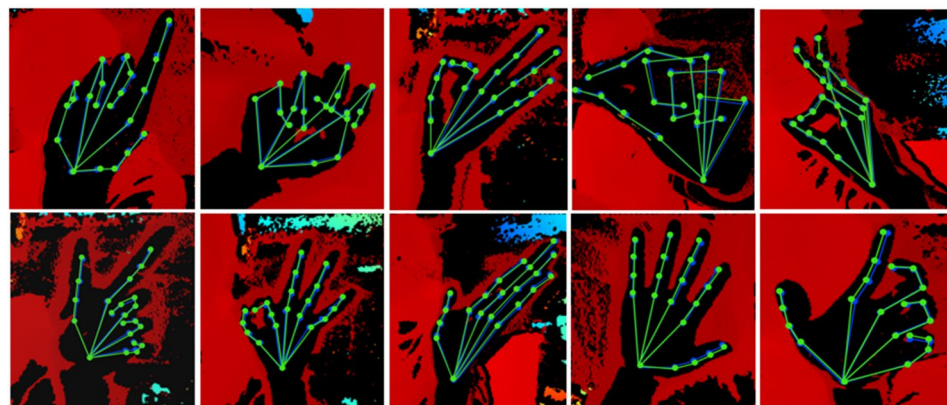


Figure 10. Visualization results of hand joint prediction on the self-collected dataset.

the ground truth annotations and blue indicates the predicted results. From the figure, it is evident that the proposed network achieves good joint position estimates across different hand gesture poses. For example, in the second image of the first row and the fourth image of the first row, there is partial occlusion between the finger joints, yet our method is still able to accurately estimate the joint positions.

In the first image of the second row, the grasping gesture causes the remaining three fingers, apart from the thumb and index finger, to appear smaller and experience insufficient lighting, but our method still accurately estimates the corresponding joint positions. These results show that, even in challenging scenarios where finger features are less distinct—such as partial occlusion or poor lighting—our method demonstrates strong robustness and accuracy, providing precise predictions of hand joint positions.

Figure 11 further demonstrates the hand gesture pose estimation results of our proposed method on the MSRA dataset, where red represents the ground truth annotations and green indicates the predicted results. The visual results clearly show that our method is capable of handling severe self-occlusion in hand poses while maintaining high accuracy and stability on the MSRA dataset.

For instance, in the third image of the first row, although the joints are not occluded, the pose results in depth inaccuracies in the metacarpophalangeal and wrist joints of the thumb and little finger due to lighting issues. Nevertheless, our method still accurately estimates the joint positions. In the first image of the second row and the second image of the second row, some joints overlap and occlude due to the nature of the hand gestures, but our method still provides accurate joint position estimates. Therefore, the MSRA dataset visualization experiment further validates the effectiveness of our proposed method in hand gesture pose estimation tasks, highlighting its strong generalization ability on publicly available datasets.

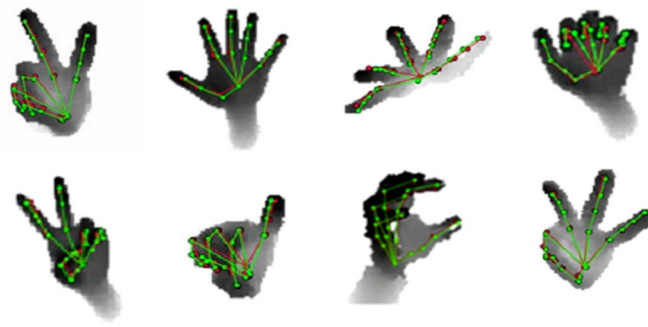


Figure 11. Visualization results of hand joint prediction on the MSRA dataset.

Conclusions

This study addresses common challenges in hand 3D pose estimation, such as the limited feature extraction and insufficient modeling of joint topological relationships. We propose a novel method for hand gesture pose estimation based on Swin Transformer and CNN. The method uses depth images of the hand region as input and utilizes Swin Transformer and CNN to capture global features and local joint features, respectively. After feature fusion, a two-dimensional Gaussian heatmap supervision is applied to refine the feature extraction process, significantly improving its accuracy and robustness.

Experimental results show that our method outperforms existing mainstream depth-based hand gesture pose estimation algorithms on both our custom dataset and the MSRA dataset. Particularly in more challenging scenarios, where finger features are less distinct or the background is more complex, our method demonstrates strong robustness and accuracy. Compared to traditional hand pose estimation methods, our approach effectively addresses the issue of insufficient modeling of long-range topological relationships between hand joints, significantly enhancing the model's adaptability across various application scenarios.

Although the method performs well in single-hand scenarios, it still faces limitations in complex environments involving hand-object interactions and severe occlusions. Future research will focus on improving algorithm performance in interactive gestures and hand-object occlusion scenarios, further strengthening the model's robustness and real-time adaptability to address more complex interactive tasks.

Data availability

The datasets analyzed during the current study are not publicly available because they were jointly generated by two research teams; however, they are available from the corresponding author on reasonable request.

Code availability

The code version corresponding to the core results of this study has been archived and is available for access through the Zenodo repository. The DOI link is: <https://doi.org/10.5281/zenodo.18649094>.

Received: 15 September 2025; Accepted: 24 February 2026

Published online: 02 March 2026

References

- Qi, J., Ma, L., Cui, Z. & Yu, Y. Computer vision-based hand gesture recognition for human-robot interaction: A review. *Complex Intell. Syst.* **10**, 1581–1606. <https://doi.org/10.1007/s40747-023-01173-6> (2024).
- Zhou, H., Wang, D., Yu, Y. & Zhang, Z. Research progress of human-computer interaction technology based on gesture recognition. *Electronics* **12**, 1–19. <https://doi.org/10.3390/electronics12132805> (2023).
- Wang, T., Zheng, P., Li, S. & Wang, L. Multimodal human-robot interaction for human? Centric smart manufacturing: A survey. *Adv. Intell. Syst.* **6**, 1–29. <https://doi.org/10.1002/aisy.202300359> (2024).
- Ding, Y., Mao, R., Du, G. & Zhang, L. Clothes-eraser: Clothing-aware controllable disentanglement for clothes-changing person re-identification. *Signal Image Video Process.* **18**, 4337–4348. <https://doi.org/10.1007/s11760-024-03076-6> (2024).
- Xie, Y., Hong, C., Zhuang, W., Liu, L. & Li, J. Hogformer: High-order graph convolution transformer for 3D human pose estimation. *Int. J. Mach. Learn. Cybern.* **16**, 599–610. <https://doi.org/10.1007/s13042-024-02262-9> (2025).
- Lee, J., Kim, J., Kim, S. H. & Choi, S.-I. Enhancing 3d hand pose estimation using Shaf: Synthetic hand dataset including a forearm. *Appl. Intell.* **54**, 9565–9578. <https://doi.org/10.1007/s10489-024-05665-x> (2024).
- Adapa, S. K., Panapana, P., Boddu, J. S., Gathram, R. B. & Atyam, M. L. N. Multi-player gaming application based on human body gesture control Syst. *Des. Appl. Intell.* **45**, 4561. https://doi.org/10.1007/978-3-031-64847-2_1 (2024).
- Oikonomidis, I., Kyriazis, N. & Argyros, A. A. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *2011 International Conference on Computer Vision* 1–8 (2011). <https://doi.org/10.1109/ICCV.2011.6126483>
- Tompson, J., Stein, M., Lecun, Y. & Perlin, K. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.* **33**, 1–10. <https://doi.org/10.1145/2629500> (2014).
- Ye, Q., Yuan, S. & Kim, T.-K. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. *Comput. Vis. ECCV* **346–361**, 2016. https://doi.org/10.1007/978-3-319-46484-8_21 (2016).
- Sun, X., Wei, Y., Liang, S., Tang, X. & Sun, J. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1–9 (2015). <https://doi.org/10.1109/CVPR.2015.7298683>
- Kun, G. et al. Challenges and solutions for vision-based hand gesture interpretation: A review. *Comput. Vis. Image Underst.* **248**, 104095. <https://doi.org/10.1016/j.cviu.2024.104095> (2024).

13. Yuanzhe, L. et al. Parameter estimation for connectome generative models: Accuracy, reliability, and a fast parameter fitting method. *NeuroImage* **270**, 119962. <https://doi.org/10.1016/j.neuroimage.2023.119962> (2023).
14. Qian, C., Sun, X., Wei, Y., Tang, X. & Sun, J. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1106–1113 (2014). <https://doi.org/10.1109/CVPR.2014.145>
15. Zhang, X. et al. Hand image understanding via deep multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* 11281–11292 (2021). <https://doi.org/10.1109/ICCV48922.2021.01109>
16. Chen, P. et al. I2uv-handnet: Image-to-UV prediction network for accurate and high-fidelity 3D hand mesh modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* 12929–12938 (2021). <https://doi.org/10.1109/ICCV48922.2021.01269>
17. Ge, L. 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 (CVPR)* 10833–10842 (2019). <https://doi.org/10.1109/CVPR.2019.01109>.
18. Zhang, X., Li, Q., Mo, H., Zhang, W. & Zheng, W. End-to-end hand mesh recovery from a monocular RGB image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* 2354–2364 (2019). <https://doi.org/10.1109/ICCV.2019.00244>
19. Fan, Z., Liu, J. & Wang, Y. Adaptive computationally efficient network for monocular 3D hand pose estimation. *Comput. Vis. ECCV* **127–144**, 2020. https://doi.org/10.1007/978-3-030-58548-8_8 (2020).
20. Pavlakos, G. et al. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9826–9836 (2024). <https://doi.org/10.1109/CVPR52733.2024.00938>.
21. Mohammad, R., Razieh, R. & Vassilis, A. Trihorn-net: A model for accurate depth-based 3d hand pose estimation. *Expert Syst. Appl.* **223**, 119922. <https://doi.org/10.1016/j.eswa.2023.119922> (2023).
22. Wen, Y. et al. Hierarchical temporal transformer for 3D hand pose estimation and action recognition from egocentric RGB videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 21243–21253 (2023). <https://doi.org/10.1109/CVPR52729.2023.02035>
23. Keskin, C., Kırç, F., Kara, Y. E. & Akarun, L. Real time hand pose estimation using depth sensors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* 1228–1234 (2011). <https://doi.org/10.1109/ICCVW.2011.6130391>
24. Sridhar, S., Mueller, F., Oulasvirta, A. & Theobalt, C. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 3213–3221 (2015). <https://doi.org/10.1109/CVPR.2015.7298941>
25. Sridhar, S., Mueller, F., Zollhöfer, M., Casas, D. & Oulasvirta, A. & Theobalt, C. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *Computer Vision - ECCV 2016* 294–310 (2016). https://doi.org/10.1007/978-3-319-46475-6_19
26. Panteleris, P., Oikonomidis, I. & Argyros, A. Using a single rgb frame for real time 3D hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* 436–445 (2018). <https://doi.org/10.1109/WACV.2018.00054>
27. Bao, L., Yang, Z., Wang, S., Bai, D. & Lee, J. Real image denoising based on multi-scale residual dense block and cascaded u-net with block-connection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* 448–449 (2020). <https://doi.org/10.1109/CVPRW50498.2020.00232>
28. Ju, M., Luo, J., Wang, Z. & Luo, H. Adaptive feature fusion with attention mechanism for multi-scale target detection. *Neural Comput. Appl.* **33**, 2769–2781. <https://doi.org/10.1007/s00521-020-05150-9> (2021).
29. Luo, Y., Xu, Z., Liu, P., Du, Y. & Guo, J. Combining fractal hourglass network and skeleton joints pairwise affinity for multi-person pose estimation. *Multimed. Tools Appl.* **78**, 7341–7363. <https://doi.org/10.1007/s11042-018-6502-7> (2019).
30. Tompson, J., Stein, M., Lecun, Y. & Perlin, K. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.* **33**, 1–10. <https://doi.org/10.1145/2629500> (2014).
31. Zhou, X., Wan, Q., Zhang, W., Xue, X. & Wei, Y. Model-based deep hand pose estimation. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence* 2421–2427 (2016). <https://doi.org/10.5555/3060832.3060960>
32. Ge, L., Liang, H., Yuan, J. & Thalmann, D. Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3593–3601 (2016). <https://doi.org/10.1109/CVPR.2016.391>
33. Oberweger, M. & Lepetit, V. Deepprior++: Improving fast and accurate 3D hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops* 585–594 (2017). <https://doi.org/10.1109/ICCVW.2017.75>
34. Xinghao, C., Guijin, W., Hengkai, G. & Cairong, Z. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing* **395**, 138–149. <https://doi.org/10.1016/j.neucom.2018.06.097> (2020).
35. Wan, C., Probst, T., Van Gool, L. & Yao, A. Dense 3D regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 5147–5156 (2018) <https://doi.org/10.1109/CVPR.2018.00540>

Author contributions

R.D. conceived and designed the study. R.D. and G.F. conducted the experiments and collected the data. R.D. implemented the model and performed the analysis. G.F. assisted in result interpretation and manuscript editing. All authors reviewed and approved the final manuscript.

Funding

This work was financially supported by Key Laboratory of Tourism Information Fusion Processing and Data Ownership Protection, Ministry of Culture and Tourism (Project NO.2024TPDP01).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026