Pattern Recognition Letters 183 (2024) 155–164

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

HandFormer: Hand pose reconstructing from a single RGB image

Zixun Jiao [a,b], Xihan Wang [a,b,*], Jingcao Li [a,b], Rongxin Gao [a,b], Miao He [a,b], Jiao Liang [a,b], Zhaoqiang Xia [c], Quanli Gao [a,b,*]

[a] State and Local Joint Engineering Research Center for Advanced Networking & Intelligent
[b] Information Services, School of Computer Science, Xi'an Polytechnic University, Xi'an, 710048, Shaanxi, China
[c] Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, China

ARTICLE INFO

Edited by: Prof. S. Sarkar

Keywords:
Hand attitude estimation
Hand attitude estimation and segmentation
Multitasking learning
Multitask progressive transformer framework
Multi-scale features

ABSTRACT

We propose a multi-task progressive Transformer framework to reconstruct hand poses from a single RGB image to address challenges such as hand occlusion hand distraction, and hand shape bias. Our proposed framework comprises three key components: the feature extraction branch, palm segmentation branch, and parameter prediction branch. The feature extraction branch initially employs the progressive Transformer to extract multi-scale features from the input image. Subsequently, these multi-scale features are fed into a multi-layer perceptron layer (MLP) for acquiring palm alignment features. We employ an efficient fusion module to enhance the parameter prediction further features to integrate the palm alignment features with the backbone features. A dense hand model is generated using a pre-computed articulated mesh deformed hand model. We evaluate the performance of our proposed method on STEREO, FreiHAND, and HO3D datasets separately. The experimental results demonstrate that our approach achieves 3D mean error metrics of 10.92 mm, 12.33 mm and 9.6 mm for the respective datasets.

1. Introduction

In recent years, there has been a significant surge of interest in the field of hand pose reconstruction, particularly in domains such as virtual reality (VR) and augmented reality (AR), where it finds extensive application [1]. Many reconstruction approaches have been explored in the last decades and deep learning technique has played a pivotal role in advancing this research field. Most of the current work focuses on deep networks, which can be roughly divided into two types: RGB based methods and RGBD (RGB+Depth) based methods.The RGBD based method simplifies the extraction of hand region depth information, but it faces some problems such as increased computational complexity, data alignment, and difficulty in annotation. Therefore, most recent research has shifted its focus to reconstructing hand posture from a single RGB image [2-5].

Unlike laboratory data, it is difficult to capture detailed hand annotations in natural scenes, and hand movements are easily affected by some factors such as object occlusion, self occlusion, lighting, and edge blur.To address these issues, some researchers used weakly supervised methods [6,7] to learn data invariance from existing annotations and then extend it to unknown data. At the same time, some studies [5,8] have introduced a hand prior model [9], which obtains parameters from extracted image features and helps to generate accurate hand poses. Some other researchers [4,10,11,39] used heatmap to regress the key-point locations and obtain more implicit information. However, it is difficult for traditional convolutional neural network (CNN) architectures to effectively capture the intricate information correlation between the occluded parts and the global information. Therefore, in order to obtain more feature information in the hand image to solve occlusion problem, the Transformer architecture [12] has been introduced into the 3D hand pose reconstruction task. Compared with the traditional CNN architecture, the Transformer can better focus on the global features without losing the local features, thus obtaining the implicit features of the occluded hand [13]. However, these methods focus on the mapping relationship from the occluded part to the global one to reconstruct the hand posture, while ignoring the implicit connection between the hand's region and the occluded area.

In this paper, we introduces HandFormer, a Transformer architecture for reconstructing 3D hand poses. This framework decouples hand pose estimation into finger estimation and palm estimation. The motivation for this decoupling is based on our observation that in occluded scenes, fingers that usually interact with objects are easily affected by occlusion,

* Corresponding author.
E-mail addresses: xihan_wang@xpu.edu.cn (X. Wang), gaoquanli@nwu.edu.cn (Q. Gao).

https://doi.org/10.1016/j.patrec.2024.05.019
Received 16 July 2023; Received in revised form 21 March 2024; Accepted 26 May 2024
Available online 27 May 2024

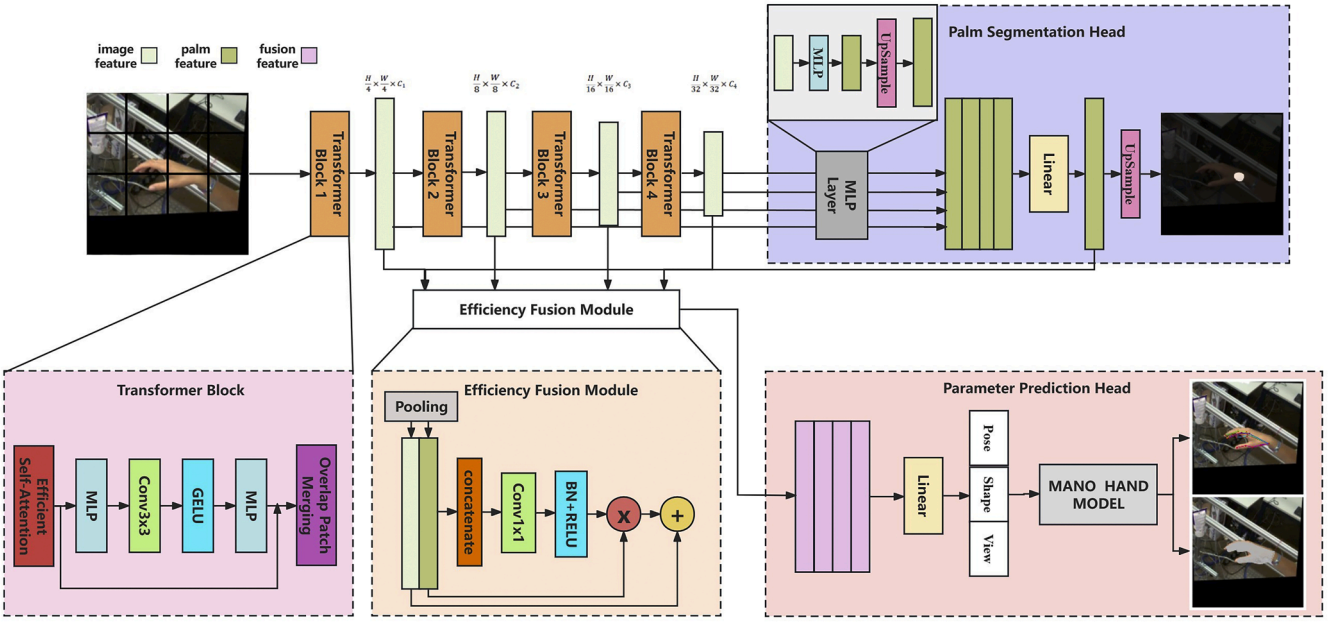

Fig. 1. In the model architecture diagram, our framework consists of four key parts: feature extraction branch, palm segmentation branch, parameter prediction branch and efficient fusion module.

while palms are primarily unaffected. Therefore, we introduced a palm segmentation module to predict the position of unobstructed hands. Firstly, Transformer is used to learn the mapping relationship between finger parts and global information. This helps to refine the final estimation of hand posture. Then, the backbone and palm segmentation features are fused through the fusion module to obtain the implicit association between the finger occlusion and palm areas. Finally, the fused features are used to predict the 3D model of the hand. Our proposed framework can potentially improve the accuracy of hand pose estimation, and the training process uses 3D data to generate the desired annotations without requiring the dataset to provide additional annotation data. Compared to other vision Transformer architectures, we use a pyramidal feature encoder [14] and design a lightweight decoder to accomplish multiple tasks using a multi-headed lightweight decoder. The main contributions of this paper are as follows:

- We propose a palm segmentation branch. Segmentation features and multilevel features are fused by an efficient fusion module to guide the generation of the final hand model. This palm branch can potentially improve the prediction of hand parameters.
- We propose that the full hand segmentation is not used. Simple hand regions obtained through keypoint can effectively constrain the hand shape effectively, adjust the overall hand position, and compute the loss function.
- We evaluate the proposed method on three publicly available datasets and demonstrate its effectiveness in various situations, including hand-object interaction, occlusion, and clutter environments.

2. Related work

In recent years, 3D hand pose estimation is mainly based on the depth neural network to achieve more accurate hand pose reconstruction. These methods can be roughly divided into two categories: RGBD based methods [7,16-23] and RGB based methods [4-6,8,10,15,24,26-29].

2.1. Hand pose reconstruction from RGBD image

Early single-hand reconstruction work used depth data to optimize the prediction of hand keypoints. Gi et al. [19] used skin detection and foreground separation algorithms to detect and segment the hand position in each RGB and depth image and then they regressed 3D hand keypoints through convolutional neural networks. Sun Y et al. [20] used a Kinect sensor to obtain depth images extracted features through double branch structure, and carried out multi-level feature fusion to regress hand pose. Zhang et al. [7] Proposed a weakly supervised confrontation learning framework for restoring human posture from depth maps.

At the same time, some researchers introduced hand models [21,22], which fits the hand model parameters by extracting the depth information of the images. Mueller et al. [23] modeled the hand as a spherical network. The collision penalty is added to the spherical network to better capture the shape of the hand and prevent model penetration. Although deep data can effectively assist in hand pose estimation, it also introduces additional computational and data annotation work. With the emergence of large-scale RGB datasets, recent research has tended to focus on hand pose reconstruction methods based on RGB image in order to reduce dependence on deep data.

2.2. Hand pose reconstruction from RGB image

Zimm et al. [15] proposed network consists of three branches for hand segmentation, prediction of 2D keypoints, and prediction of 3D keypoints, respectively. The network initially learns the hand feature information in the image through residual networks and then constrains the location of the hand pose by segmenting the hand region. This work laid the foundation for subsequent deep learning-based hand pose estimation research. Some RGB based method [4,6] learned the locations of keypoints of the hand by the weakly supervised approach, which only requires to mark the regions where the 2D keypoints are located. To obtain dense 3D hand shapes, other researchers [5,8,10] have used the popular hand model Mano [9] for acceptable hand pose reconstruction. Bouk et al. [5] proposed the concept of hand region to ensure that the network can learn the occluded part while restricting the hand model fit to a reasonable range. Zhou et al. [24] used segmentation networks to obtain hand regions and extract significant hand features to guide the network in learning local and non local relationships. Barbhuiya [26] obtains hand posture by strengthening the attention of the hand region.

In order to better extract hand features, Transformer architecture is introduced into the hand estimation task. Through this architecture, the mapping relationship between global and local features of the hand can be captured [14,25]. Others [27-29] have proposed using a hybrid feature attention network to obtain multi-scale hand features to refine edge details, or using Transformer to learn local and non local relationships [40] in images through a multi head attention mechanism to ensure that the network preserves occluded hand features.

However, the hand pose reconstruction from a single RGB image still faces problems such as the inability to accurately reconstruct the hand occluded by objects and the lack of accurate annotation information. Therefore, this paper proposes a new framework, which first uses the Transformer as the backbone network to extract the global relationship mapping and then uses the efficient fusion module to integrate the palm segmentation features layer by layer to guide the parameter prediction module to reconstruct the hand posture.

3. Proposed method

In this section, we present our designed model, a multi-task progressive Transformer framework, as shown in Fig. 1. Firstly, we describe our network's overall framework, consisting of the feature extraction, palm segmentation, and parameter prediction branches. Secondly, we explain the loss function required for training.

3.1. Overall network architecture

Pyramid Transformer structure in the progressive backbone feature extraction branch. This structure effectively extracts multi-scale semantic information and reduces the computational workload. The features extracted from the backbone network are sent to two decoder branches: the palm segmentation branch and the parameter prediction branch. In addition, we designed an efficient fusion module. In this module, the segmented features of the palm and the trunk features are fused layer by layer, so that the parameter prediction branch can capture the semantic features of palm alignment.

3.1.1. Progressive feature extraction branch

In this paper, we adopt the pyramid feature structure of PVT as the backbone of the model. Considering the image continuity, we treat every 4×4 pixels as a patch, which is then fed into the Transformer Block. In each Transformer Block, features are first computed using an efficient multi-headed attention module. Next, position features between vectors are obtained through variable-length position encoding. Finally, convolution operations combine different patches to capture more interaction information.

3.1.1.1. Efficient self-attention. Due to the computational overload of the original multi-headed attention mechanism, each head Q, K, and V has the same dimension $N \times C$, where $N = H \times W$ is the length of the sequence, and the self-attention estimate is:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_{head}}}\right)V \tag{1}$$

The time complexity in Eq. (1) is $O(N^2)$, and in the transformer block we introduce a simplification factor R using the sequence simplification method proposed in Wang et al. [14] which can effectively reduce the time complexity of the self-focus mechanism, as shown in Eq. (2,3).

$$K' = Reshape\left(\frac{N}{R}, C{\cdot}R\right)(K) \tag{2}$$

$$K = Linear(C{\cdot}R, C)(K') \tag{3}$$

We first scale K by a scaling factor R, to $\left(\frac{N}{R}, C{\cdot}R\right)$ a vector of size, and then pass the linear layer $Linear(C{\cdot}R, C)(\cdot)$ changing it into a $\left(\frac{N}{R}, C\right)$ size feature output, which reduces the time complexity of the self-attentive mechanism in Eq. (1) from $O(N^2)$ to $O\left(\frac{N^2}{R}\right)$ In our experiments, we set different scaling factors R in each stage, which are [64, 16, 4, 1], respectively.

3.1.1.2. Positional encoding of indefinite length. We employ the position encoding method of indefinite length to obtain more accurate position information and ensure the consistency of multi-headed attentional features [30]. This helps the model learn the semantic relevance of each patch being in the correct position.

3.1.1.3. Overlapped patch merging. To ensure that the information between each patch can be fused and interacted with, we use a convolution operation to interact information across patches. For this purpose, we define Kn, S and P, where Kn is the kernel size, S is the step size between two adjacent patches and P is the padding size. In our experimental setup, we initially configured the parameters for Block1 as $Kn = 7$, $S = 4$, and $P = 3$. Subsequently, we adjusted the parameters for the remaining Blocks to $Kn = 3$, $S = 2$, and $P = 1$. Overlapping patches are merged through convolution operations of these parameters to generate features of the same size as the non-overlapping process.

With each Transformer Block we are able to get multi-level semantic features, exactly given an input image with a resolution of $H \times W \times C$, we will get at each Block $\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i$ $i \in \{1, 2, 3, 4\}$ output features of size where C_{i+1} is larger than C_i.

3.1.2. Palm segmentation head

Using hand segmentation maps to constrain hand pose estimation is a widely adopted approach in the field [5,31]. However, most existing methods [31] rely on hand mask annotations provided by the dataset. In our approach, we propose to restrict the hand shape to the approximate hand region without the need for a complete hand segmentation. Therefore, we use the 2D keypoint positions in the labeled data to obtain a simple hand region mask (as shown in Fig. 3), thus restricting the hand shape to that region via Eq. (9).

Some approaches [5,31] using segmentation networks ignore the fact that hand-object interactions can lead to discontinuities in the segmented region. For this reason, we propose the palm segmentation branch to obtain hand pose information from the unobstructed hand portion. Compared to other methods [5,31] that use heatmap information to localize the hand pose, segmenting only the palm does not introduce too much background noisy information, forcing the network to focus on the palm portion, which in turn enables the anchoring of the hand shape. Specifically, we scale the components obtained in each block of the progressive backbone feature extraction branch to the size of $\frac{H}{4} \times \frac{W}{4} \times$ 256 features and then fuse them by the concatenate operation, and finally reduce them to the original map size by a superficial linear layer and upsample to obtain the segmentation map.

3.1.3. Efficient fusion module

In the efficient fusion module, we aim to fuse the feature information of each Block in the backbone feature extraction stage with the palm position feature information in the segmentation branch to achieve the overall hand alignment. Firstly, we resize each Block's output features and the segmentation branch's output features to the same size for stitching. Then, we use the RELU activation function and the convolution operation of the batch normalization layer to obtain the fusion vector weight, so that the network can focus on the palm region to enhance the local characteristics of the segmented branches. Finally, we add the trunk feature and the palm segmentation feature layer by layer, and get the fusion feature with the size of $\frac{H}{32} \times \frac{W}{32} \times$ 256. This design allows us to utilize both the global semantic information and the local

palm position information better to capture the structure and features of the hand.

3.1.4. Parameter prediction head

The parameter prediction head is to reconstruct the hand pose's mesh surface to achieve this goal. We use the most popular MANO parametric hand [9]. It is an articulated hand network generated by differentiable functions that combine the parameters controlling the hand shape β and pose θ as inputs, as follows:

$$M(\beta,\theta) = W(T(\beta,\theta), J(\beta), \theta, \omega) \tag{4}$$

where W is a linear blending mask function [32] and is applied to a triangular mesh T that contains a kinematic tree with $K = 16$ joints. J denotes the joint position and it is learned as a sparse linear regression volume from the mesh vertices, where ω is the blending weight. To reduce artifacts of the linear blending mask, such as over-smoothed output and mesh collapse around the joints, the hand mesh T is obtained by deforming the mean mesh $\overline{T}$ using shape and pose corrected blending shapes S_n and P_n, respectively, as follows:

$$T(\beta,\theta) = \overline{T} + \sum_{n=1}^{|\beta|} \beta_n S_n + \sum_{n=1}^{9K} (R_n(\theta) - R_n(\theta^*))P_n \tag{5}$$

Also, in order to project the generated hand model back into the original image, we use a weak perspective model that can be simply and easily mapped back to the original image for training purposes without knowing the internal parameters of the camera, as follows:

$$X' = S\prod(RJ(\beta,\theta)) + t \tag{6}$$

$$Y' = S\prod(RM(\beta,\theta)) + t \tag{7}$$

where the rotation matrix $R \in SO(3)$, the scaling factor S, and the translation parameter t are $\prod$ given to represent the orthographic projection.

In the parameter prediction head, we simply splice the outputs of the layers of the efficient fusion module and feed them to the linear layer for parameter prediction. Finally, we obtain a set of parameters containing the camera View=$\{R, t, s\}$, shape β and pose θ of the hand. Finally, we generate a hand model from these parameters and project it back to the original image.

3.2. Loss function

Our model is trained using a combination of multiple loss functions, which include the 2D keypoint loss ($Loss_{2d}$), the 3D keypoint loss ($Loss_{3d}$), the hand region coverage loss ($Loss_{cover}$), the palm segmentation loss ($Loss_{mask}$), and the model parameter regularization loss ($Loss_{reg}$). These loss functions are jointly optimized to train the model effectively. The overall loss function is formulated in Eq. (8), where weighting factors ($\omega_{3d} = 10^2$, $\omega_{cover} = 10^2$, $\omega_{mask} = 10^1$ and $\omega_{reg} = 10^1$) are used to balance the contributions of each loss component. By incorporating these loss functions, we aim to capture various aspects of the hand pose estimation task and achieve comprehensive training of our model.

$$Loss = Loss_{2d} + \omega_{3d}Loss_{3d} + \omega_{cover}Loss_{cover} + \omega_{mask}Loss_{mask} + \omega_{reg}Loss_{reg} \tag{8}$$

2D keypoint loss, we utilize the L1 loss to quantify the discrepancy between the model-predicted 2D keypoint and the ground truth 2D keypoint annotations.

3D keypoint loss, when 3D keypoint annotations are available, we use L2 loss to measure the gap between model predictions and real-world annotations.

Hand area coverage loss, we employ this loss function to confine the hand representation within the designated hand region. By computing the intersection between the predicted hand mask and the hand region mask, we can quantify the number of overlapping pixels. This loss penalizes pixels where the predicted hand mask extends beyond the boundaries of the hand region, ensuring that the model focuses on accurately capturing the hand within its designated area.

$$Loss_{cover} = 1 - \frac{1}{N}\sum_i H(Y_i') \tag{9}$$

In Eq. (9), H represents the hand region mask, and $H(u)$ is defined as 1 when pixel u exists within the hand region H, and 0 otherwise.

Palm segmentation loss, in real-world scenarios, occlusion and variations in illumination caused by object interactions primarily affect regions other than the palm. However, the alignment of the palm region significantly influences the overall alignment of the hand shape within the hand region. Therefore, we train the network to extract palm region features, specifically targeting the problem of correcting hand shape alignment. By focusing on the palm region, which is less affected by occlusion and illumination changes, we aim to improve the overall alignment accuracy of the hand shape within the hand region.

$$Loss_{mask} = L_{Focal} + L_{Dice} \tag{10}$$

$$L_{Focal}(P_t) = -(1 - P_t)^{\gamma} log(P_t) \tag{11}$$

$$L_{Dice} = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \tag{12}$$

Due to the small percentage of the target region in palm segmentation, Focal Loss is introduced to weigh the easy and difficult-to-classify samples to alleviate the category imbalance problem, where P_t denotes the prediction confidence of the model for the samples. γ is the focusing parameter, which controls the degree of weight adjustment for quickly and complicated classified samples. In Eq. (12), X denotes the actual label, while Y denotes the predicted label.

Regularization loss, this loss acts on the parameter branch to constrain the predicted parameters to reduce the physically reasonable hand reconstruction magnitude and reduce hand mesh distortion, where θ and β denote the attitude and shape parameters in Eq. (4), $\alpha_\beta = 10^4$ is used as a weighting factor:

$$Loss_{reg} = \| \theta \| + \alpha_\beta \| \beta \| \tag{13}$$

4. Experiment

4.1. Implementation details

The model's training process comprises two distinct phases: pre-training and formal training. Initially, during the pre-training stage, we employ a learning rate of 1e-3, a batch size 64, and conduct training for 100 epochs. Subsequently, in the formal training phase, we maintain the identical learning rate and batch size used in the pre-training, while continuing to train for 100 epochs. The training is executed on a server equipped with a CPU15 vCPU AMD processor and an RTX A5000 GPU boasting 24GB of memory. The server serves as the platform for conducting subsequent tests as well.

4.2. Data set and evaluation indicators

To ensure comprehensive training across all branches of the network, a combination of synthetic and publicly available datasets was employed. The synthetic dataset consists of 2959 non-private images sourced from the internet via a web crawler. To promote background diversity during data generation, we randomly crop fixed sized windows from different background images.

We utilized the same hand models as the branches responsible for predicting the parameters to facilitate the convergence of hand model parameters within an acceptable range. These hand models were

Fig. 2. Synthetic data set.

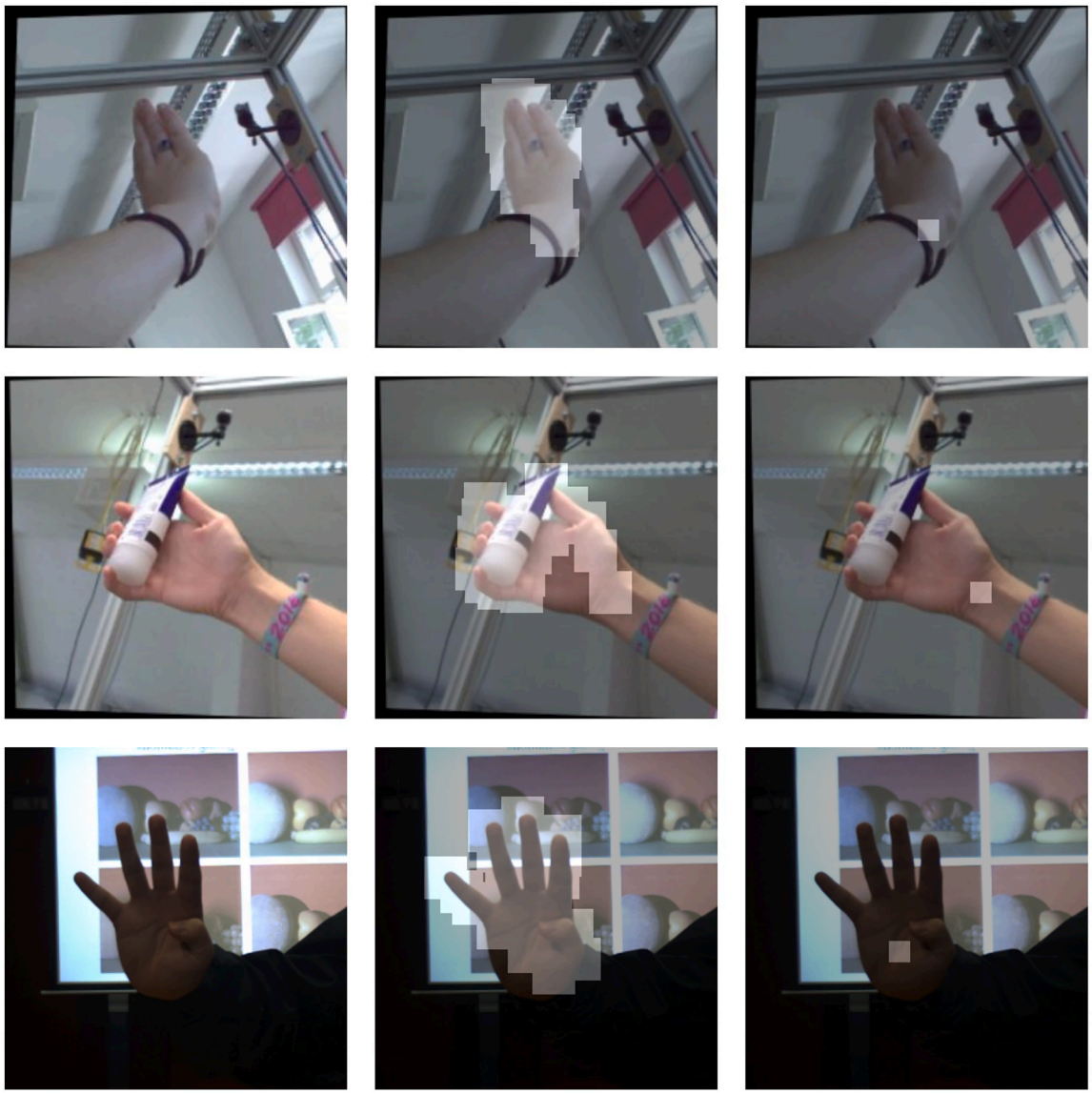

Fig. 3. Original data set (left), hand area mask (center), and palm segment mask (right).

generated for different poses and then projected onto the acquired images with various backgrounds using weak perspective projection, as illustrated in Fig. 2.

During the pre-training phase, the images underwent data augmentation techniques, including color transformations, Gaussian noise, and random masking. However, changes that would alter the hand

Table 1
Average distance and AUC to ground-truth for DATASETS.

STEREO			
	2D distance(px)	3D distance(mm)	AUC(%)
Zimm el.at	–	23.21	94.6
Bouk el.at	–	**10.46**	98.4
ours	**10.06**	10.92	**99.7**
FreiHAND			
Zimm el.at	–	12.47	84.8
Bouk el.at	–	13.0	85.5
ours	**12.58**	**12.33**	**94.2**

morphology, such as scaling and flipping, were deliberately avoided. This was done to ensure the robustness of the generated hand models and to exhibit generalization capabilities across different poses and contexts.

Our training dataset comprises three primary datasets: HO3D [10], FreiHAND [33], and STEREO [2]. Overall, our dataset contains about 120 K training images. Among them, the data from STEREO and FreiHAND have only joint annotations, while the data from HO3D include MANO parameters and are provided with an online test system for validation.

We utilize the keypoint locations for all train datasets to generate the hand region's mask image. Additionally, we create the mask image for the segmented palm based on the palm location. For a detailed understanding of the processing method and an example, please refer to Fig. 3.

To evaluate the performance of our proposed model on RGB images and assess its effectiveness across different scenarios, we perform comparisons and ablation experiments with recent methods on test sets from three publicly available datasets. These datasets encompass images exhibiting occlusion, low lighting, object interaction, and blurring conditions. On the STEREO and FreiHAND datasets we evaluate using PCK and L1 mean error distance; on the HO3D dataset, we evaluated using an online evaluation system, reporting the average joint error, average mesh error, and F-score based on Procrustes align [10].

4.3. Comparison to competing methods

In our research, we use keypoint annotations and partial MANO parameter annotations. Remarkably, our approach demonstrates superior performance and accuracy compared to other deep learning-based methods [2,5,15,32,33] in terms of both accuracy and performance. This holds even when compared to strategies that incorporate additional depth information or employ recently popular heat map inputs [10,11,19] during training and those that utilize camera parameters during the evaluation phase.

On the STEREO dataset, we compare the results of our paper with recent state-of-the-art methods [2,5,15,34], showcasing the 2D and 3D average error distances in Table 1. Additionally, in Fig. 4, we present a comparison with recent results obtained using deep learning methods (e.g., Bouk et al. [5], Zimm et al. [15]) as well as methods not relying on deep learning (e.g., PSO, Panteleris et al. [34], Zhang et al. [2]).

From the figure, it is evident that the growth rate of the 3DPCK metric starts to slow down when the error reaches 25 mm. The accuracy of various methods tends to stabilize, with the machine learning PSO method achieving only 54 % accuracy at a 25 mm error, while other deep learning-based methods [2,5,15,34] exceed 70 %.

However, challenges remain in aligning the palm region and addressing interaction edge errors in these methods. In contrast, our proposed method achieves a PCK metric of 98.65 % at a 20 mm error, showcasing an improvement of nearly 14 percentage points compared to the deep network proposed by Zimm et al. [33]. Similarly, it surpasses the recent method proposed by Bouk [5], with an improvement of 2.46 % compared to their results.

Furthermore, we also measure the mean error metric on the STEREO dataset, which includes real-world interaction images with potential issues such as shadows and blurred edges in the hand regions. However, our method demonstrates excellent generalization in such cases, as evidenced by the overall mean error metric. Specifically, the average error is 10.06 pixels for 2D and 10.92 mm for 3D.

Fig. 4. PCK for STEREO and FreiHAND compares the model proposed in this paper and the recent competing methods [5,33] concerning PCK on the FreiHAND dataset. As shown in Table 1, the area under the curve (AUC) of the model in this paper has reached 94.2 %, nearly 10 % higher than that of the competitive methods.

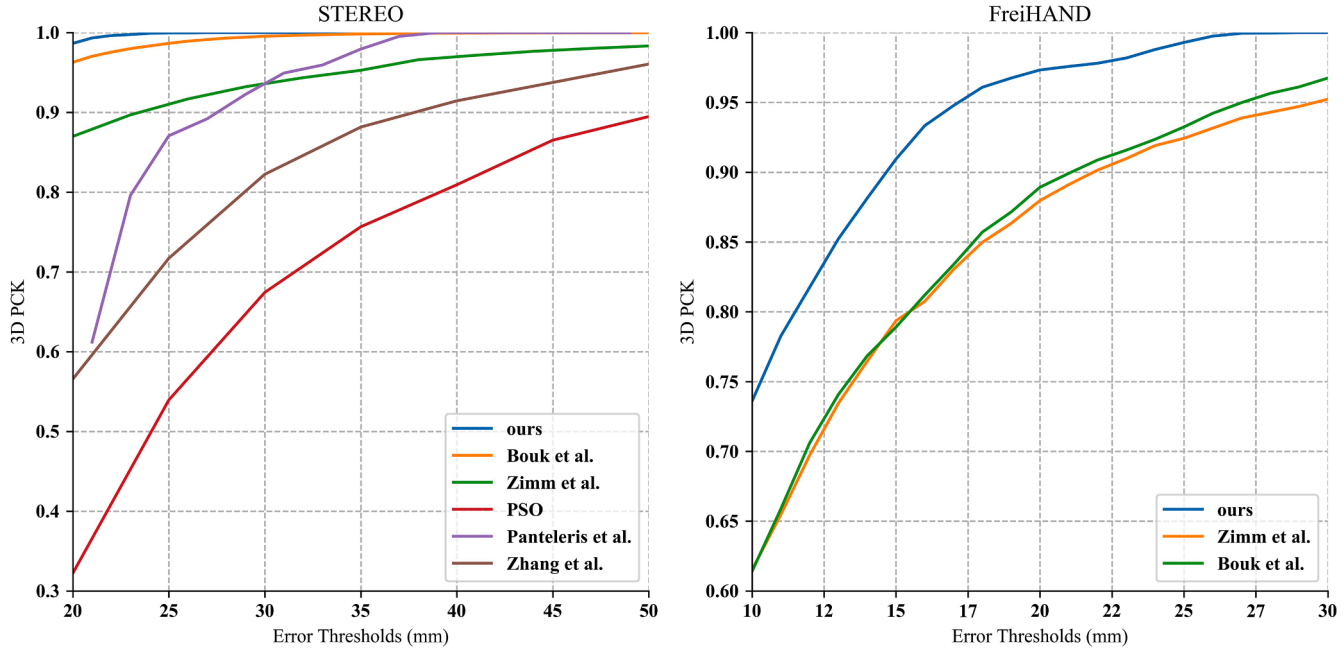

Fig. 4. PCK for STEREO and FreiHAND.

Table 2
Results of HO3D comparison experiments (* indicates that the results are from Methods [10]).

Methods	Joint	Mesh	F@5	F@15
*Pose2Mesh [36]	12.5	12.7	44.1	90.9
*Hasson et al. [37]	11.4	11.4	42.8	93.2
*I2L-MeshNet [11]	11.2	13.9	40.9	93.2
*Hampali et al. [38]	10.7	10.6	50.6	94.2
*METRO [13]	10.4	11.1	48.4	94.6
*Liu et al. [35]	10.2	9.8	**52.9**	95.0
*Keypoint Transformer [10]	11.0	11.3	44.4	93.5
Ours	**9.6**	**9.6**	52.1	**95.6**

On the FreiHAND dataset, we examine the 3DPCK metric with an error range of 10 to 30 mm . The results demonstrate that our method rapidly increases accuracy, consistently maintaining a PCK metric above 95 % when the error exceeds 21 mm . Across the entire error range, our proposed method significantly outperforms the approaches proposed by Zimm and Bouk on this dataset. Specifically, when the error reaches 20 mm , our process exhibits an improvement of approximately 9 % and 7 % relative to Zimm's and Bouk's methods, respectively.

Furthermore, we measure the average 2D and 3D errors, revealing an average error of 12.58 pixels for 2D and 12.33 mm for 3D. These results demonstrate that our method can also achieve favorable outcomes on untrained test sets.

Table 2 shows the metrics comparison with recent methods on the HO3D dataset [10] with data from an online review site. We evaluate the hand reconstruction results by joint error, mesh error, F@5, and F@15 metrics. We outperform recent methods [10] by about 13 % in standard and mask errors, and including palm, features enables better hand alignment in occlusion situations. We outperform Liu et al. [35] in the F@15 metric but slightly underperform in F@5. This may be because the palm portion is constrained by the palm features so that most of the hand regions overlap, but the finger portion is heavily occluded. There is a partial offset of the fingers, resulting in a more significant error in the F@5 case.

In Fig. 5, we present the result plots of each branch of the model proposed in this paper on the test set. The figure showcases four images from left to right: the input image, the output image of the palm segmentation branch, the 2D keypoints, and the mesh image.

Observing these images shows that our proposed palm segmentation branch accurately identifies the region of the hand's root nodal point.

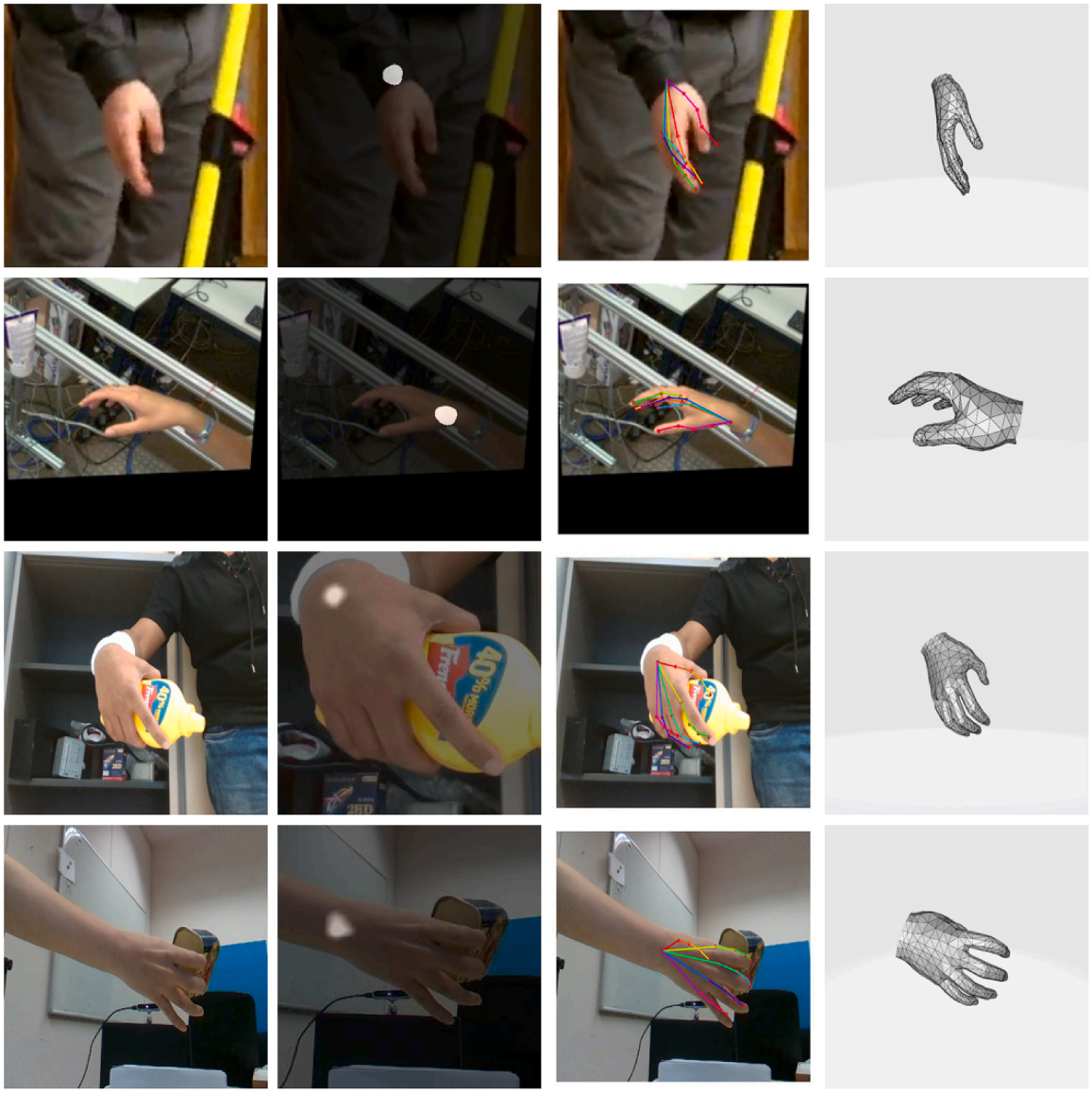

Fig. 5. Test results of the method in this article (from left to right: original image, palm segmentation, 2D keypoint prediction, hand mesh mask).

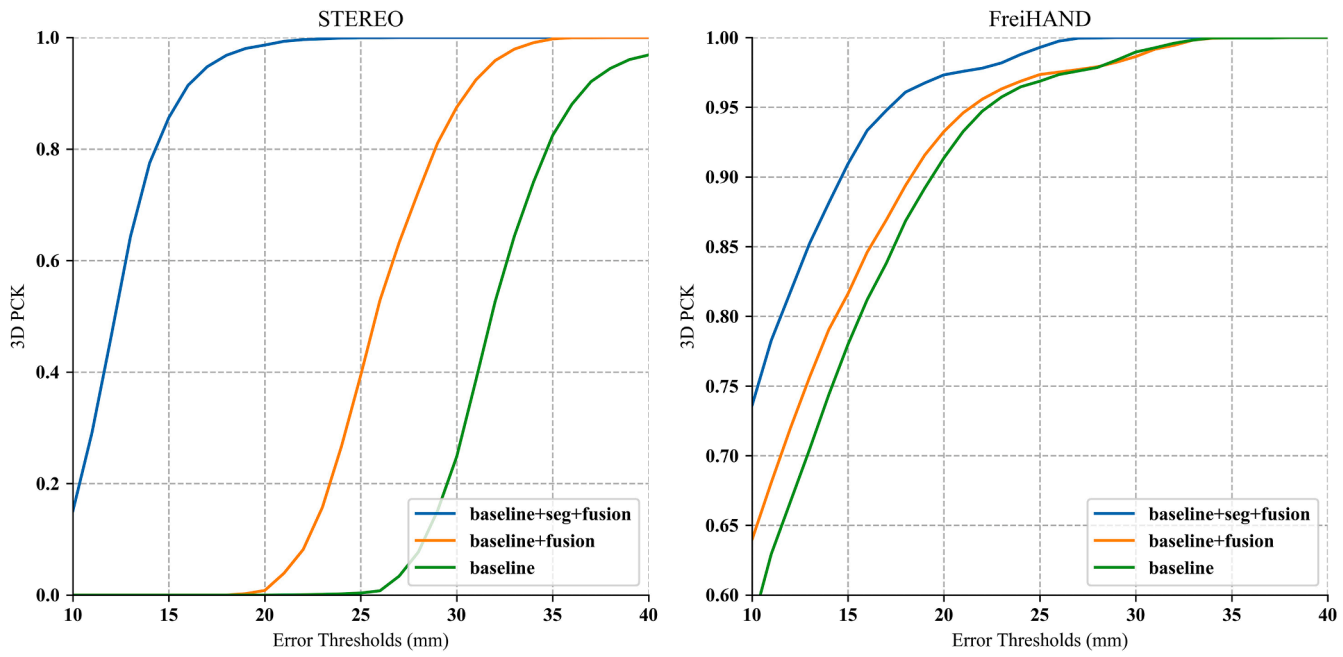

Fig. 6. Ablation experiment for STEREO and FreiHAND.

Table 3
Results of STEREO and FreiHAND ablation experiment.

STEREO			
	baseline	baseline+fusion	baseline+seg+fusion
2D distance	66.75	47.58	**10.06**
3D distance	32.26	26.11	**10.92**
FreiHAND			
	baseline	baseline+fusion	baseline+seg+fusion
2D distance	37.23	31.05	**12.58**
3D distance	17.79	13.49	**12.33**

Table 4
Results of HO3D ablation experiment.

Methods	Joint	Mesh	F@5	F@15
Baseline	17.5	17.3	29.9	81.8
Baseline+Fusion	10.6	10.5	48.5	94.0
Baseline+Fusion+Heatmap(palm)	9.9	9.7	**52.3**	95.1
Baseline+Fusion+Heatmap(21 keypoints)	9.7	9.7	51.9	95.3
Baseline+Fusion+Seg(ours)	**9.6**	**9.6**	52.1	**95.6**

This branch also adjusts the overall pose of the hand to ensure alignment from the root, thereby enabling fine-tuning of the finger part to align with most of the hand's fitted region. These results affirm the effectiveness of our approach in accurately segmenting the palm region and aligning the hand pose for improved performance.

4.4. Ablation experiment

To validate the effectiveness of the proposed palm segmentation branch and efficient fusion module, we conducted ablation experiments. These experiments aimed to verify that the palm segmentation branch successfully learns hand palm alignment features. At the same time, the efficient fusion module effectively combines these alignment features to expedite the alignment of 2D keypoints on the hand and enhance accuracy.

We opted to perform ablation experiments using the HO3D [10], FreiHAND [33], and STEREO [2] datasets mentioned earlier. The validation set comprises real-world images depicting hand-object interactions, encompassing challenges such as hand self-obscuration, occlusion due to handheld objects, lighting variations, and low-resolution scenarios.

In the ablation experiments, we established three models: the base model, the model with the addition of the efficient fusion module, and the model with the reserve of both the palm segmentation branch and the efficient fusion module. For FreiHAND and STEREO, we present the PCK metrics for the ablation experiments in Fig. 6, while Table 3 displays the average distance error. The results of the HO3D-based ablation experiments are demonstrated in Table 4.

Upon visual inspection of the images, the impact of incorporating an efficient fusion module alongside the baseline model becomes apparent. The efficient fusion module effectively integrates features from various levels based on their significance, resulting in a more comprehensive perception field. Consequently, it facilitates extracting non-local hand information from multi-scale features, enhancing the model's accuracy.

Furthermore, the introduction of a palm segmentation module further enhances the model. This module learns palm-specific features within hand images, ensuring alignment across most hand regions. The palm features are then combined with the multi-scale hand features by calculating weights for the palm features at each scale using the efficient fusion module. By doing so, the network captures palm-specific characteristics and emphasizes global hand edge information, ensuring overall hand alignment while fine-tuning the alignment of individual finger components.

With this design approach, our model maximizes the utilization of multi-scale and palm features, improving hand pose estimation accuracy. It ensures hand alignment and fine-tuning, further enhancing the model's performance.

The average 2D error of the baseline model in STEREO is 66.75 pixels. When the efficient fusion module was introduced, the error was reduced to 47.58 pixels. With the additional palm segmentation module, the error further decreased to only 1/6 of the baseline model, reaching 10.06 pixels. Similarly, the average 3D error decreased from 26.11 mm to 10.92 mm , approximately 1/3 of the baseline model's error.

In the FreiHAND dataset, the baseline model initially performed slightly better, with an average 2D error of 37.23 pixels. After

incorporating the fusion module, the error slightly decreased to 31.05 pixels. However, when the palm segmentation module was added, the error was directly halved to a final average 2D error of 12.58 pixels. The average 3D error of the baseline model started at 17.79 mm , decreased by 4 mm after adding the fusion module, and finally reduced to 12.33 mm after introducing the palm segmentation module.

For the HO3D dataset, the hand pose reconstruction is affected because the image features contain a more significant proportion of noisy background features. In baseline+fusion, we fuse the image features layer by layer to restore the hand pose, and this progressive fusion makes up for the missing hand detail information in the deeper features and improves the metrics by about 40 %. Meanwhile, in order to verify the necessity of using palm segmentation, we use palm heatmap features and 21 keypoint heatmap features as fusion information to restore hand pose, respectively. From the table, we can see that in the keypoint error and mesh error metrics, the heatmap method is lower than the method using palm segmentation as fusion features due to the presence of some noisy information, and only the palm-only heatmap method is slightly higher by 0.2 % in the F@5 metrics. In baseline+fusion+seg, the palm segmentation feature can limit the attention to the palm region, and the layer-by-layer fusion with the image features excludes the influence of the interference features, which strengthens the weight of the palm features, and then affects the overall pose reconstruction effect, and the final key point error and mesh error reaches 9.6 mm.

The test results on three datasets indicate that the efficient fusion module successfully captures multi-scale non-local information. The network learns the mapping relationship between high-level and low-level features through multi-level fusion, effectively addressing the occlusion problem. On the other hand, the palm segmentation module directly learns palm features and aligns the overall hand pose using these features. Finally, the non-local information learned through the joint fusion module is utilized to fine-tune the alignment of the occluded finger parts. These results demonstrate the effectiveness of our proposed modules in improving accuracy and addressing occlusion challenges.

5. Conclusion

We introduce a novel approach for estimating hand pose from a single RGB image using the Transformer architecture. Leveraging Transformers's global information-gathering capability, we capture the implicit semantics of hand images and employ semantic segmentation branches to localize the palm keypoints. By combining the palm alignment features with the parameter prediction branches, we achieve comprehensive alignment of the MANO hand model. During the training process, we extend it based on limited annotations to help the network learn more information while achieving good results on data with full supervision.

CRediT authorship contribution statement

Zixun Jiao: Writing – review & editing, Writing – original draft, Validation, Software. **Xihan Wang:** Writing – review & editing, Supervision, Investigation, Funding acquisition, Data curation. **Jingcao Li:** Investigation, Validation, Writing – review & editing. **Rongxin Gao:** Writing – review & editing, Investigation. **Miao He:** Investigation, Writing – review & editing. **Jiao Liang:** Investigation, Writing – review & editing. **Zhaoqiang Xia:** Writing – review & editing, Investigation. **Quanli Gao:** Supervision, Resources, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgements

This work was supported by the National Natural Science Foundation of China, Shaanxi Provincial Key Industry Innovation Chain Program, the 2024 public digital cultural service research project of the national public cultural development center of the Ministry of culture and tourism, and the scientific research program of Shaanxi Provincial Department of Education [grant numbers 62072362 and 12101479, 2020ZDLGY07-05, GGSZWHFW2024-017, 23jp060].

References

[1] Markus Höll, et al., Efficient physics-based implementation for realistic hand-object interaction in virtual reality, in: IEEE Conference on Virtual Reality and 3D User Interfaces, 2018.
[2] Jiawei Zhang, et al., 3D Hand Pose Tracking and Estimation Using Stereo Matching, CVPR, 2016.
[3] Mueller, et al., GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB, CVPR, 2017.
[4] Iqbal, et al., Hand Pose Estimation Via Latent 2.5D Heatmap Regression, ECCV, 2018.
[5] Boukhayma, et al., 3D Hand Shape and Pose From Images in the Wild, CVPR, 2019.
[6] Adrian Spurr, et al., Weakly Supervised 3D Hand Pose Estimation Via Biomechanical Constraints, ECCV, 2020.
[7] Zihao Zhang, et al., Weakly Supervised Adversarial Learning For 3D Human Pose Estimation from Point Clouds, TVCG, 2020.
[8] Ge, et al., 3d Hand Shape and Pose Estimation from a Single Rgb Image, CVPR, 2019.
[9] Javier Romero, et al., Embodied hands: Modeling and Capturing Hands and Bodies Together, SIGGRAPH Asia, 2017.
[10] S. Hampali, et al., Key-point transformer: Solving joint Identification in Challenging Hands and Object Interactions For Accurate 3d Pose Estimation, CVPR, 2022.
[11] Gyeongsik Moon, et al., I2L-MeshNet:Image-to-lixel Prediction Network For Accurate 3D Hu-Man Pose and Mesh Estimation from a Single RGB Image, ECCV, 2020.
[12] Dosovitskiy, et al., An Image is Worth 16x16 Words: Transformers for Image Recognition At Scale, ICLR, 2021.
[13] Lin, et al., End-to-End Human Pose and Mesh Reconstruction With Transformers, CVPR, 2021.
[14] Wang, et al., Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions, ICCV, 2021.
[15] C. Zimmermann, et al., Learning to Estimate 3D Hand Pose from Single RGB Images, ICCV, 2017.
[16] M. Wu, et al., Hand pose estimation in object-interaction based on deep learning for virtual reality applications, J. Vis. Commun. Image Represent. (2020).
[17] H. Xing, et al., Learning dynamic relationship between joints for 3D hand pose estimation from single depth map, J. Vis. Commun. Image Represent. (2023).
[18] Andrea D'Eusanio, et al., Depth-based 3D human pose refinement: evaluating the refinet framework, Pattern. Recognit. Lett. (2023).
[19] Gi, et al., Real Time 3D Pose Estimation of Both Human Hands via RGB-Depth Camera and Deep Convolutional Neural Networks, In BME 7 (2020).
[20] Y. Sun, et al., Gesture recognition algorithm based on multi-scale feature fusion in RGB-D images, IET Image Process. (2023).
[21] Tkach, et al., Sphere-meshes for real-time hand modeling and tracking, ACM Trans. Graph. (2016).
[22] Khamis, et al., Learning an efficient model of hand shape variation from depth images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.
[23] Mueller, et al., Real-time pose and shape reconstruction of two interacting hands with a single depth camera, ACM Trans. Graph. (ToG) (2019).
[24] W. Zhou, A lightweight hand gesture recognition in complex backgrounds, Displays (2022).
[25] Enze Xie, et al., SegFormer: simple and Efficient Design for Semantic Segmentation with Transformers, NeurIPS (2021).
[26] Barbhuiya, et al., Gesture recognition from RGB images using convolutional neural network-attention based system, Concurr. Comput.: Pract. Exp. (2022).
[27] Bhaumik, et al., Hyfinet: hybrid feature attention network for hand gesture recognition, Multimed. Tools. Appl. (2023).
[28] C. Jiang, et al., A2J-Transformer: Anchor-to-Joint Transformer Network for 3D Interacting Hand Pose Estimation from a Single RGB Image, CVPR, 2023.
[29] Huang, et al., Hand-transformer: Non-Autoregressive Structured Modeling For 3d Hand Pose Estimation, ECCV, 2020.
[30] Md Amirul Islam, et al., How Much Position Information Do Convolutional Neural Networks Encode, ICLR, 2020.

[31] X. Zhang, et al., Hand image understanding via deep multi-task learning. CVPR, 2021.
[32] Ladislav Kavan, et al., Spherical blend skinning: a real-time deformation of articulated models, in: Proceedings of the 2005 symposium on Interactive 3D graphics and games (I3D '05), 2005.
[33] Zimmermann, et al., FreiHAND: A Dataset For Markerless Capture of Hand Pose and Shape From Single RGB Images, ICCV, 2019.
[34] P. Panteleris, et al., Using a Single Rgb Frame For Real Time 3d Hand Pose Estimation in the Wild, WACV, 2018.
[35] Shaowei Liu, et al., Semi-supervised 3D Hand-Object Poses Estimation With Interactions in Time, CVPR, 2021.
[36] Hongsuk Choi, et al., Pose2Mesh: Graph convolutional Network For 3D Human Pose and Mesh Recovery from a 2D Human Pose, ECCV, 2020.
[37] Yana Hasson, et al., Lever- aging photometric Consistency Over Time For Sparsely Supervised Hand-Object Reconstruction, CVPR, 2020.
[38] Shreyas Hampali, et al., HOnnotate: A method For 3d Annotation of Hand and Object Poses, CVPR, 2020.
[39] Park, et al., HandOccNet: Occlusion-Robust 3D Hand Mesh Estimation Network, CVPR, 2022.
[40] Z. Xia, et al., Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions, IEEE Trans. Multimed. (2022).